Exhibit 2

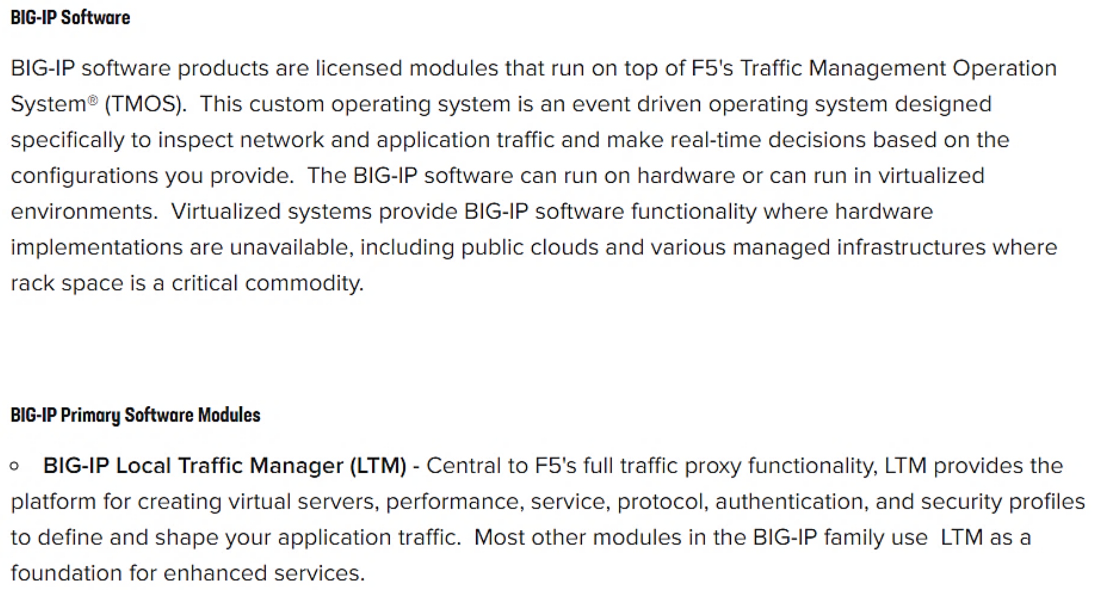
US7860000B2	Specification Support	BIG-IP Local Traffic Manager (LTM) (The Accused Product)
1pre. A method comprising: **1a.** estimating, by a processor, an activity factor for a priority class based on a provided bit rate for the priority class and a guaranteed bit rate of the priority class, wherein the activity factor defines a percentage of time that a user is active; and	When performing admission control in a system, that includes HSDPA, for **a new user of a certain priority and with certain requirements,** which can be expressed in GBR and maximum delay constraints, the admission control function of the RNC needs to know what the required power is of the existing users. The required power depends on the QoS requirements of the existing users. The Node B provides a required power attribute per priority class (SPI). This required power is defined as "the minimum necessary power for a given priority class to meet the Guaranteed Bit	The accused product practices a method for estimating, by a processor, an activity factor for a priority class based on a provided bit rate for the priority class and a guaranteed bit rate of the priority class, wherein the activity factor defines a percentage of time that a user is active. BIG-IP Local Traffic Manager (LTM) is a F5 Networks' BIG-IP Primary Software Module. LTM provides the platform for creating virtual servers, performance, service, protocol, authentication, and security profiles to define and shape the application traffic. See Fig 1. **Citation 1: About BIG-IP Local Traffic Manager (LTM)**  **BIG-IP Software** BIG-IP software products are licensed modules that run on top of F5's Traffic Management Operation System® (TMOS). This custom operating system is an event driven operating system designed specifically to inspect network and application traffic and make real-time decisions based on the configurations you provide. The BIG-IP software can run on hardware or can run in virtualized environments. Virtualized systems provide BIG-IP software functionality where hardware implementations are unavailable, including public clouds and various managed infrastructures where rack space is a critical commodity. **BIG-IP Primary Software Modules** ○ **BIG-IP Local Traffic Manager (LTM)** - Central to F5's full traffic proxy functionality, LTM provides the platform for creating virtual servers, performance, service, protocol, authentication, and security profiles to define and shape your application traffic. Most other modules in the BIG-IP family use LTM as a foundation for enhanced services. Fig 1 Source: https://devcentral.f5.com/s/articles/what-is-big-ip-24596, Page 2, Last accessed June 19, 2020, Exhibit D

Exhibit 2

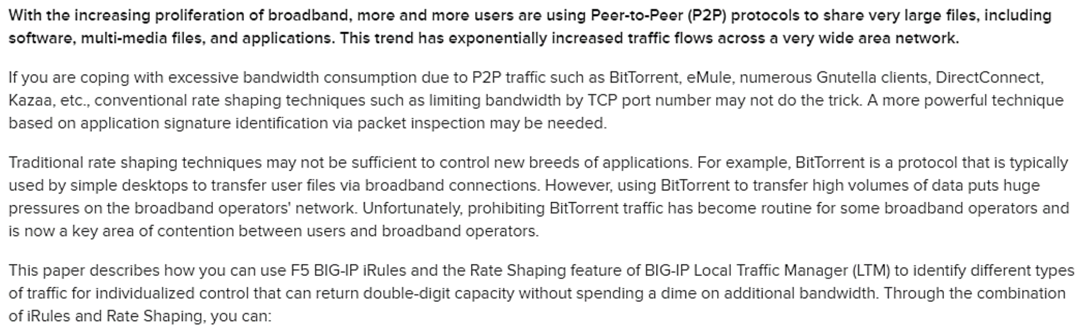
	Rate for all the established HS-DSCH connections belonging to this priority class', and assumes 100% user activity. [Col. 2, Line 7-18] The AF follows the user behavior and defines the **percentage of the time that the user (the UE associated with the user) is active**. The effective value may vary significantly among users. [Col. 2, Line 22-25] **The estimation of the activity factor may, for example, be used to estimate the used power per priority class, which in turn may be used for QoS-based admission control to adjust the HS-DSCH required power.** That is, the AC uses the	The users connected to a network managed by BIG-IP use Peer-to-Peer (P2P) protocols to share very large files, including software, multi-media files, and applications which results in excessive bandwidth consumption. A traffic class is used to classify traffic according to a set of criteria that you define, such as source and destination IP addresses. To deliver optimal application performance, F5 BIG-IP iRules and the Rate Shaping feature of BIG-IP Local Traffic Manager (LTM) (*i.e.,* the processor) identifies different types of traffic (*i.e.,* classes) to allocates more bandwidth for higher priority applications (*i.e.*, associated to priority class) by providing flexible bandwidth limits, bandwidth borrowing, and traffic queuing features. The bandwidth usage of any type of traffic can be controlled. See Fig 2 *to* Fig 4. **Citation 2: BIG-IP iRules and Rate Shaping feature**  **With the increasing proliferation of broadband, more and more users are using Peer-to-Peer (P2P) protocols to share very large files, including software, multi-media files, and applications. This trend has exponentially increased traffic flows across a very wide area network.** If you are coping with excessive bandwidth consumption due to P2P traffic such as BitTorrent, eMule, numerous Gnutella clients, DirectConnect, Kazaa, etc., conventional rate shaping techniques such as limiting bandwidth by TCP port number may not do the trick. A more powerful technique based on application signature identification via packet inspection may be needed. Traditional rate shaping techniques may not be sufficient to control new breeds of applications. For example, BitTorrent is a protocol that is typically used by simple desktops to transfer user files via broadband connections. However, using BitTorrent to transfer high volumes of data puts huge pressures on the broadband operators' network. Unfortunately, prohibiting BitTorrent traffic has become routine for some broadband operators and is now a key area of contention between users and broadband operators. This paper describes how you can use F5 BIG-IP iRules and the Rate Shaping feature of BIG-IP Local Traffic Manager (LTM) to identify different types of traffic for individualized control that can return double-digit capacity without spending a dime on additional bandwidth. Through the combination of iRules and Rate Shaping, you can: Fig 2 Source: https://www.f5.com/services/resources/white-papers/bandwidth-management-for-peer-to-peer-applications, Page 1, Last accessed June 19, 2020, Exhibit B

Exhibit 2

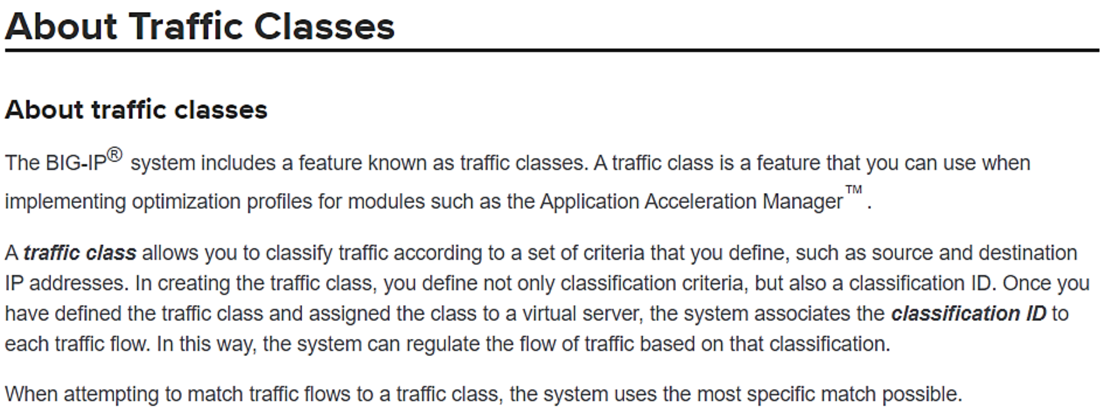
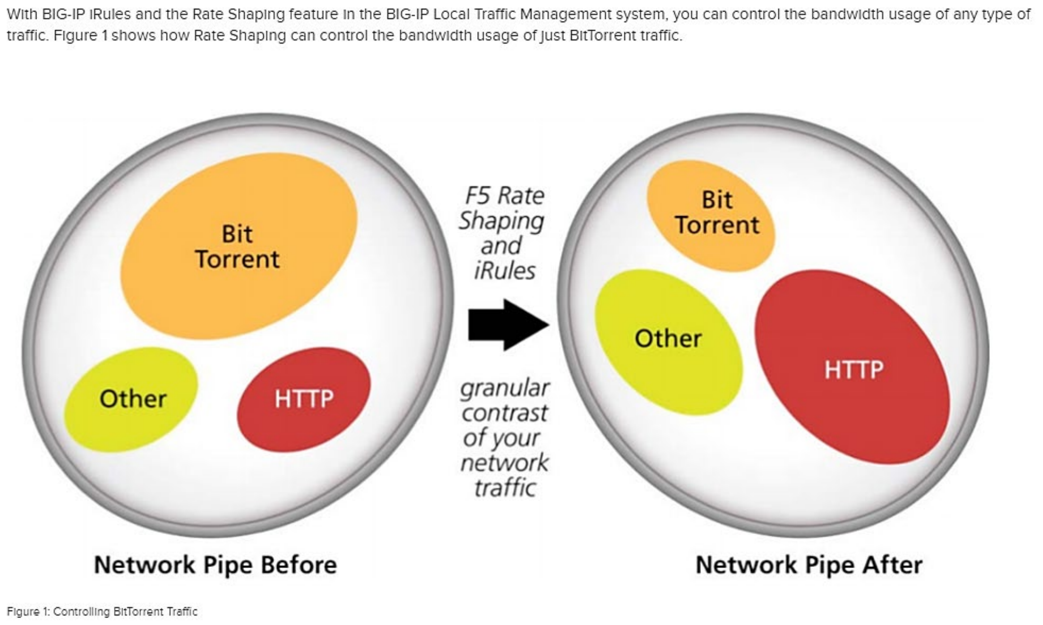
	adjusted required power for AC decisions. [Col. 4, Line 7-11]	**Citation 3: Traffic classes**  **About Traffic Classes** **About traffic classes** The BIG-IP® system includes a feature known as traffic classes. A traffic class is a feature that you can use when implementing optimization profiles for modules such as the Application Acceleration Manager™ . A ***traffic class*** allows you to classify traffic according to a set of criteria that you define, such as source and destination IP addresses. In creating the traffic class, you define not only classification criteria, but also a classification ID. Once you have defined the traffic class and assigned the class to a virtual server, the system associates the ***classification ID*** to each traffic flow. In this way, the system can regulate the flow of traffic based on that classification. When attempting to match traffic flows to a traffic class, the system uses the most specific match possible. Fig 3 Source: Bandwidth Management for Peer-to-Peer Applications, Page 1, Last accessed June 19, 2020, Exhibit A

Exhibit 2

Citation 4: Rate Shaping and iRule in BIG-IP LTM



With BIG-IP iRules and the Rate Shaping feature in the BIG-IP Local Traffic Management system, you can control the bandwidth usage of any type of traffic. Figure 1 shows how Rate Shaping can control the bandwidth usage of just BitTorrent traffic.

Figure 1: Controlling BitTorrent Traffic

Fig 4

Source: https://www.f5.com/services/resources/white-papers/bandwidth-management-for-peer-to-peer-applications, Page 2, Last accessed June 19, 2020, Exhibit B

An iRule is a feature within the BIG-IP LTM system that can be used to manage network traffic passing through an F5 device. Through the combination of iRules and Rate Shaping, critical applications are not impacted by non-priority traffic and optimal application performance can be delivered by allocating more bandwidth for higher priority applications. See Fig 5.

Exhibit 2

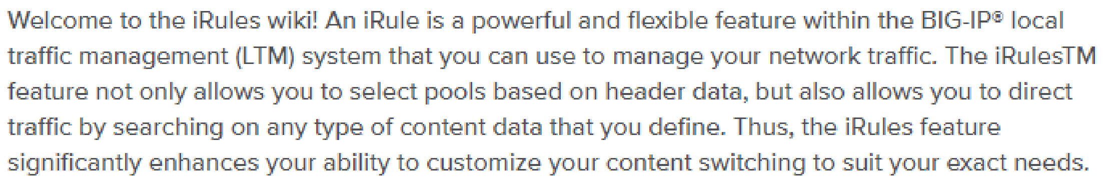
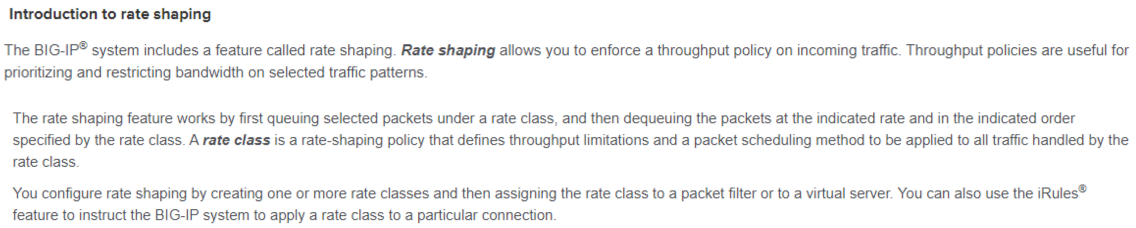
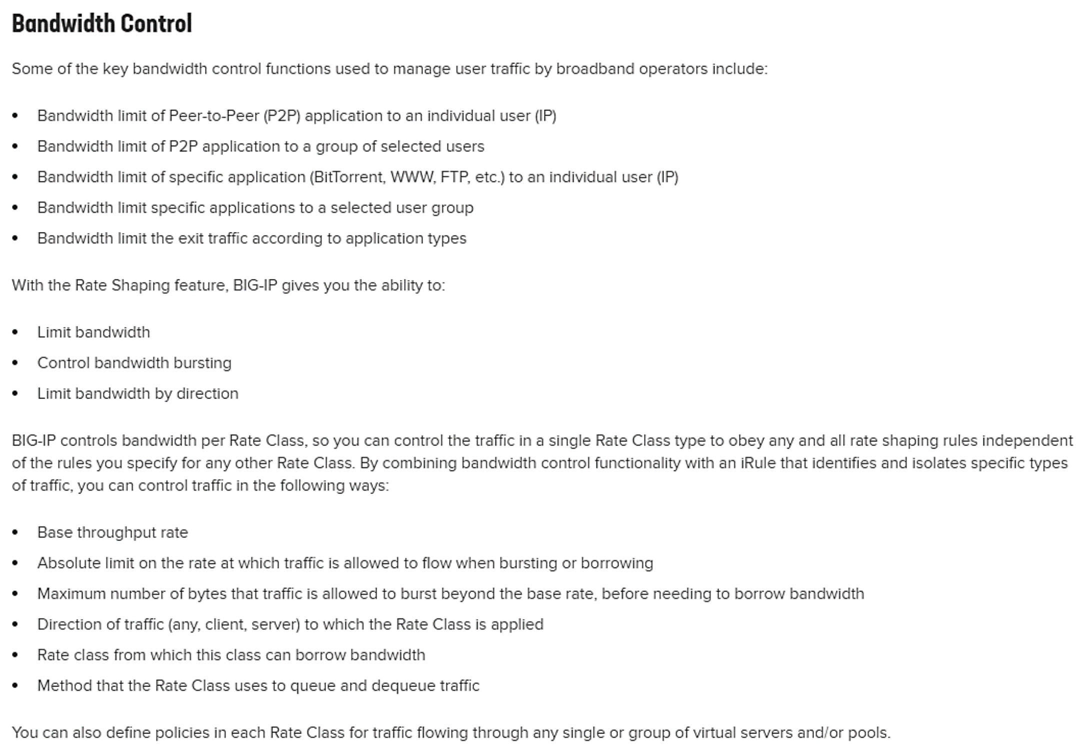
		Citation 5 : iRule feature  Welcome to the iRules wiki! An iRule is a powerful and flexible feature within the BIG-IP® local traffic management (LTM) system that you can use to manage your network traffic. The iRulesTM feature not only allows you to select pools based on header data, but also allows you to direct traffic by searching on any type of content data that you define. Thus, the iRules feature significantly enhances your ability to customize your content switching to suit your exact needs. Fig 5 Source: https://clouddocs.f5.com/api/irules/, Page 1, Last accessed September 2, 2020, Exhibit E Rate Shaping is a feature in the BIG-IP system that allows the user to enforce a throughput policy on incoming traffic for prioritizing and restricting bandwidth on selected patterns. See Fig 6. **Citation 6 : Rate Shaping feature**  **Introduction to rate shaping** The BIG-IP® system includes a feature called rate shaping. ***Rate shaping*** allows you to enforce a throughput policy on incoming traffic. Throughput policies are useful for prioritizing and restricting bandwidth on selected traffic patterns. The rate shaping feature works by first queuing selected packets under a rate class, and then dequeuing the packets at the indicated rate and in the indicated order specified by the rate class. A ***rate class*** is a rate-shaping policy that defines throughput limitations and a packet scheduling method to be applied to all traffic handled by the rate class. You configure rate shaping by creating one or more rate classes and then assigning the rate class to a packet filter or to a virtual server. You can also use the iRules® feature to instruct the BIG-IP system to apply a rate class to a particular connection. Fig 6 Source: https://techdocs.f5.com/kb/en-us/products/big-ip-aam/manuals/product/aam-concepts-11-5-0/4.html, Page 1, Last accessed September 2, 2020, Exhibit F

Exhibit 2

Bandwidth control functions are used to manage user traffic by broadband operators. The Bandwidth limit of specific applications and rate classes can be controlled by setting the ceiling rate. Also, a critical application or traffic class (priority class) can be allocated more bandwidth by providing non-important applications lesser traffic. Also, the limit of available bandwidth can be set in a range. The available bandwidth for this class will not fall below the set limit of bandwidth (guaranteed bit rate). See Fig 7 and Fig 8.

Citation 7: Bandwidth control



Bandwidth Control

Some of the key bandwidth control functions used to manage user traffic by broadband operators include:

- Bandwidth limit of Peer-to-Peer (P2P) application to an individual user (IP)
- Bandwidth limit of P2P application to a group of selected users
- Bandwidth limit of specific application (BitTorrent, WWW, FTP, etc.) to an individual user (IP)
- Bandwidth limit specific applications to a selected user group
- Bandwidth limit the exit traffic according to application types

With the Rate Shaping feature, BIG-IP gives you the ability to:

- Limit bandwidth
- Control bandwidth bursting
- Limit bandwidth by direction

BIG-IP controls bandwidth per Rate Class, so you can control the traffic in a single Rate Class type to obey any and all rate shaping rules independent of the rules you specify for any other Rate Class. By combining bandwidth control functionality with an iRule that identifies and isolates specific types of traffic, you can control traffic in the following ways:

- Base throughput rate
- Absolute limit on the rate at which traffic is allowed to flow when bursting or borrowing
- Maximum number of bytes that traffic is allowed to burst beyond the base rate, before needing to borrow bandwidth
- Direction of traffic (any, client, server) to which the Rate Class is applied
- Rate class from which this class can borrow bandwidth
- Method that the Rate Class uses to queue and dequeue traffic

You can also define policies in each Rate Class for traffic flowing through any single or group of virtual servers and/or pools.

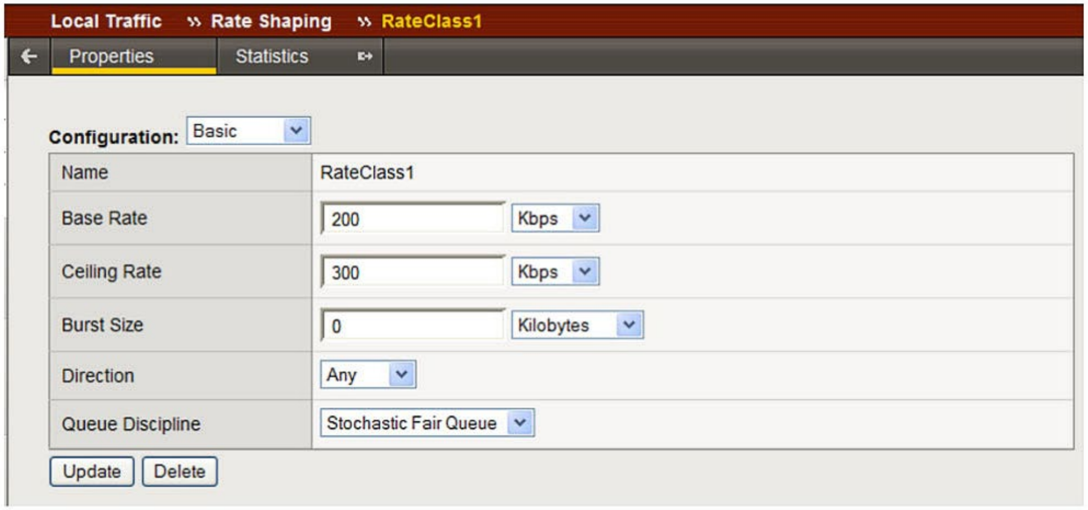
Fig 7

Exhibit 2

Source: https://www.f5.com/services/resources/white-papers/bandwidth-management-for-peer-to-peer-applications, Page 3, Last accessed June 19, 2020, Exhibit B

Citation 8: Ceiling rate

The following example shows the interface and properties for a basic rate class.



Configuration: Basic

Name	RateClass1
Base Rate	200 Kbps
Ceiling Rate	300 Kbps
Burst Size	0 Kilobytes
Direction	Any
Queue Discipline	Stochastic Fair Queue

Load balancing, monitoring, and prioritizing OCSP services.

Fig 8

Source: https://www.f5.com/services/resources/white-papers/bandwidth-management-for-peer-to-peer-applications, Page 4, Last accessed June 19, 2020, Exhibit B

The bandwidth limit for HTTP can be set for a duration of time. E.g., see Fig 9 and Fig 10.

Exhibit 2

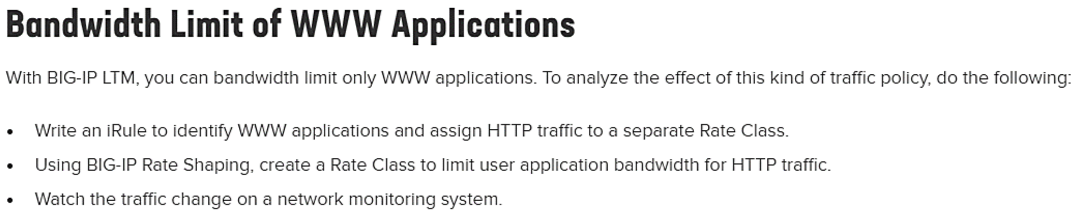
		Citation 9: Bandwidth limit  **Bandwidth Limit of WWW Applications** With BIG-IP LTM, you can bandwidth limit only WWW applications. To analyze the effect of this kind of traffic policy, do the following: • Write an iRule to identify WWW applications and assign HTTP traffic to a separate Rate Class. • Using BIG-IP Rate Shaping, create a Rate Class to limit user application bandwidth for HTTP traffic. • Watch the traffic change on a network monitoring system. Fig 9 Source: https://www.f5.com/services/resources/white-papers/bandwidth-management-for-peer-to-peer-applications, Page 7, Last accessed June 19, 2020, Exhibit B

Exhibit 2

Citation 10: HTTP Traffic Limit

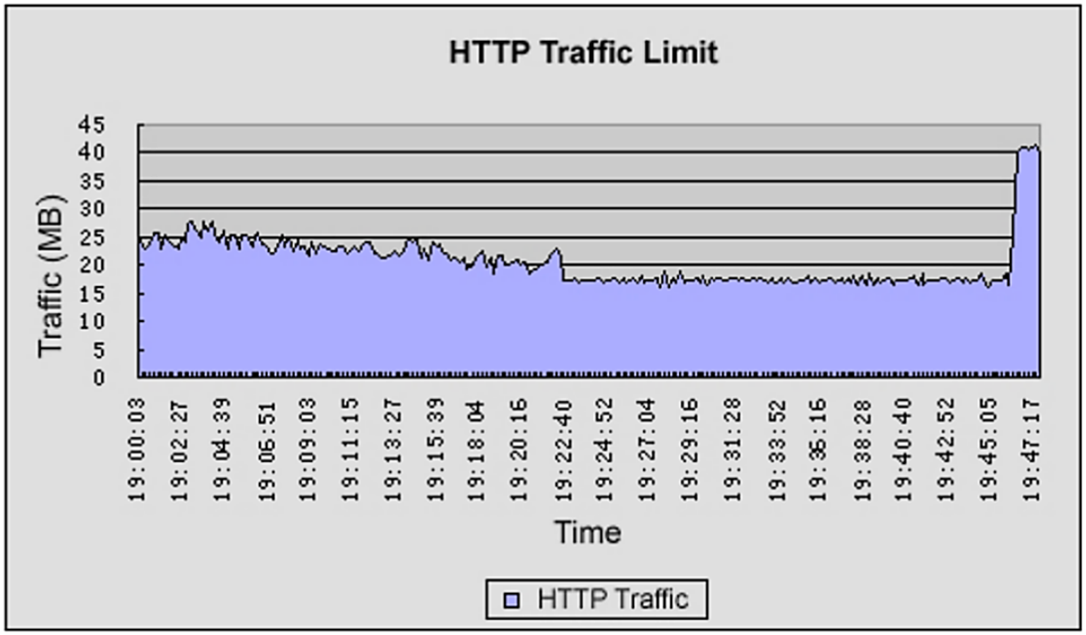


Figure 6: HTTP Traffic Limit

When HTTP is limited, pages open more slowly and HTTP application performance decreases. Once the bandwidth limit is lifted, users' HTTP traffic rises to around 40MB/s when the limit is canceled at around 19:46. HTTP traffic was limited to 18M/s while the traffic policy is in force to prevent degrading users' non-HTTP applications during the test.

Fig 10

Exhibit 2

		Source: https://www.f5.com/services/resources/white-papers/bandwidth-management-for-peer-to-peer-applications, Page 7, Last accessed June 19, 2020, Exhibit B Monitoring software can analyse real-time Bandwidth Consumption (provided bit rate) of Specific Application or traffic class. See Fig 11. **Citation 11: Bandwidth Consumption**

Exhibit 2

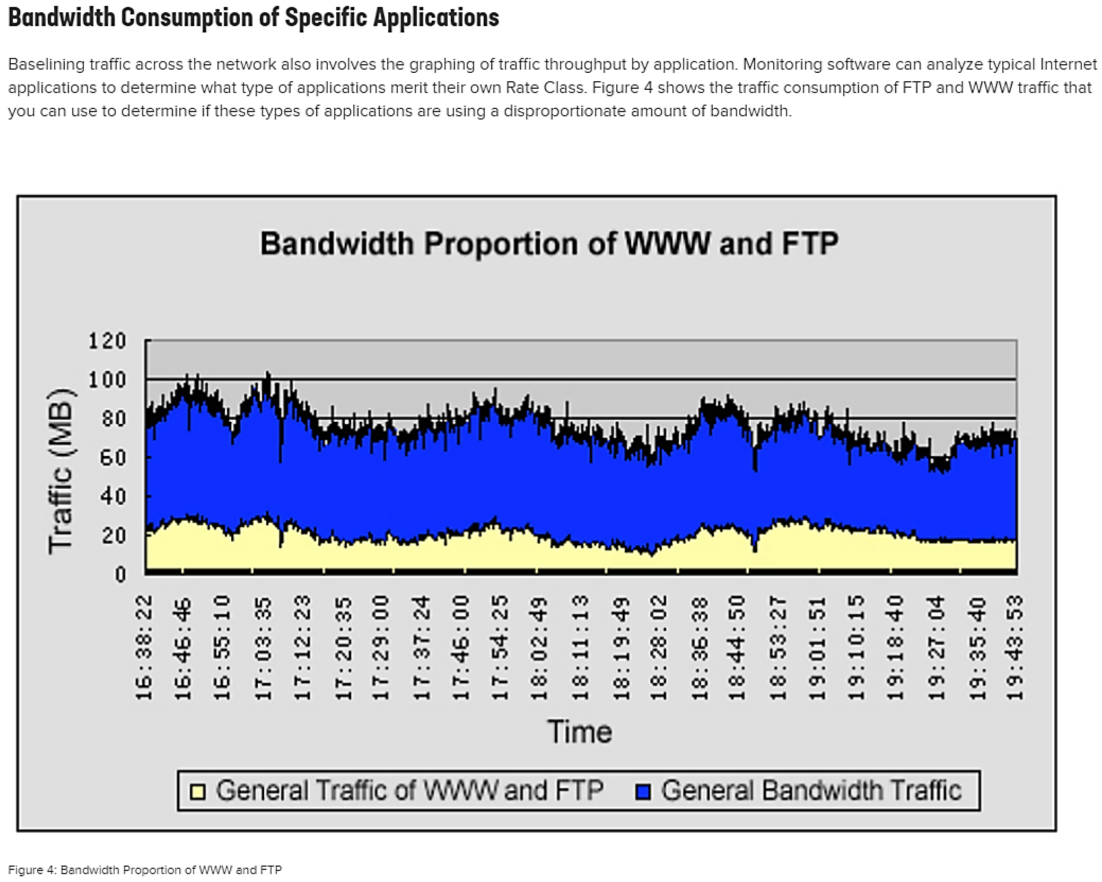
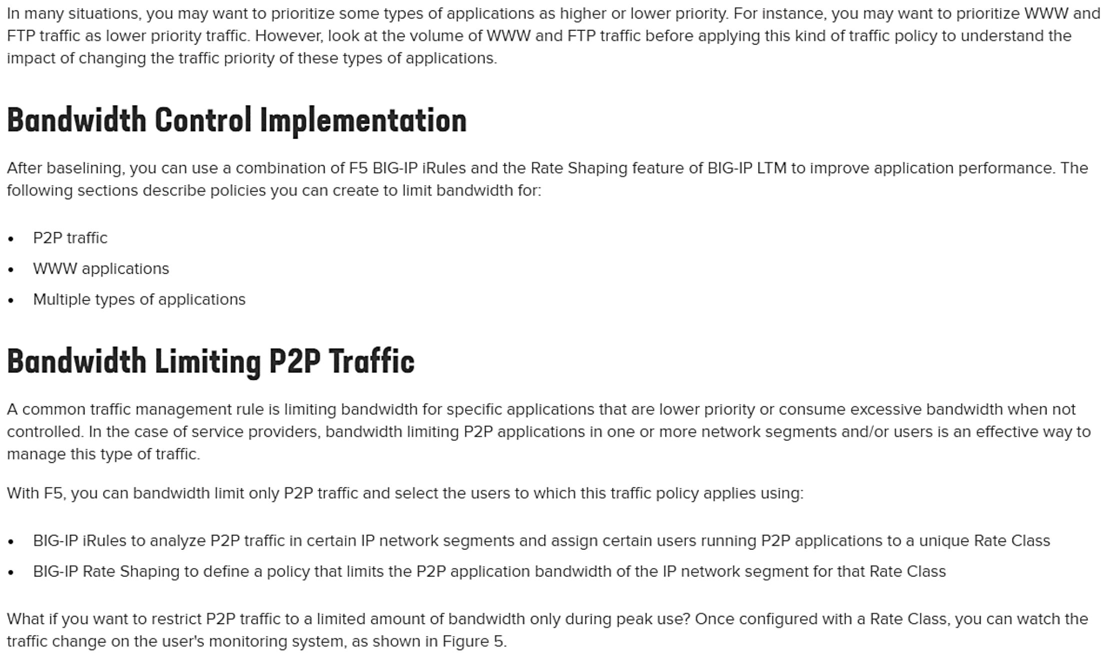
		 **Bandwidth Consumption of Specific Applications** Baselining traffic across the network also involves the graphing of traffic throughput by application. Monitoring software can analyze typical Internet applications to determine what type of applications merit their own Rate Class. Figure 4 shows the traffic consumption of FTP and WWW traffic that you can use to determine if these types of applications are using a disproportionate amount of bandwidth. Bandwidth Proportion of WWW and FTP Traffic (MB) 120 100 80 60 40 20 0 16:38:22 16:46:46 16:55:10 17:03:35 17:12:23 17:20:35 17:29:00 17:37:24 17:46:00 17:54:25 18:02:49 18:11:13 18:19:49 18:28:02 18:36:38 18:44:50 18:53:27 19:01:51 19:10:15 19:18:40 19:27:04 19:35:40 19:43:53 Time □ General Traffic of WWW and FTP ■ General Bandwidth Traffic Figure 4: Bandwidth Proportion of WWW and FTP Fig 11 Source: https://www.f5.com/services/resources/white-papers/bandwidth-management-for-peer-to-peer-applications, Page 6, Last accessed June 19, 2020, Exhibit B Also, BIG-IP Local Traffic Manager (LTM) gives the feature for restricting the bandwidth for a Specific Application or traffic class. See Fig 12 and Fig 13.

Exhibit 2

Citation 12: Bandwidth Control implementation



In many situations, you may want to prioritize some types of applications as higher or lower priority. For instance, you may want to prioritize WWW and FTP traffic as lower priority traffic. However, look at the volume of WWW and FTP traffic before applying this kind of traffic policy to understand the impact of changing the traffic priority of these types of applications.

Bandwidth Control Implementation

After baselining, you can use a combination of F5 BIG-IP iRules and the Rate Shaping feature of BIG-IP LTM to improve application performance. The following sections describe policies you can create to limit bandwidth for:

- P2P traffic
- WWW applications
- Multiple types of applications

Bandwidth Limiting P2P Traffic

A common traffic management rule is limiting bandwidth for specific applications that are lower priority or consume excessive bandwidth when not controlled. In the case of service providers, bandwidth limiting P2P applications in one or more network segments and/or users is an effective way to manage this type of traffic.

With F5, you can bandwidth limit only P2P traffic and select the users to which this traffic policy applies using:

- BIG-IP iRules to analyze P2P traffic in certain IP network segments and assign certain users running P2P applications to a unique Rate Class
- BIG-IP Rate Shaping to define a policy that limits the P2P application bandwidth of the IP network segment for that Rate Class

What if you want to restrict P2P traffic to a limited amount of bandwidth only during peak use? Once configured with a Rate Class, you can watch the traffic change on the user's monitoring system, as shown in Figure 5.

Fig 12

Source: https://www.f5.com/services/resources/white-papers/bandwidth-management-for-peer-to-peer-applications, Page 6, Last accessed June 19, 2020, Exhibit B

Exhibit 2

Citation 13: Bandwidth limit

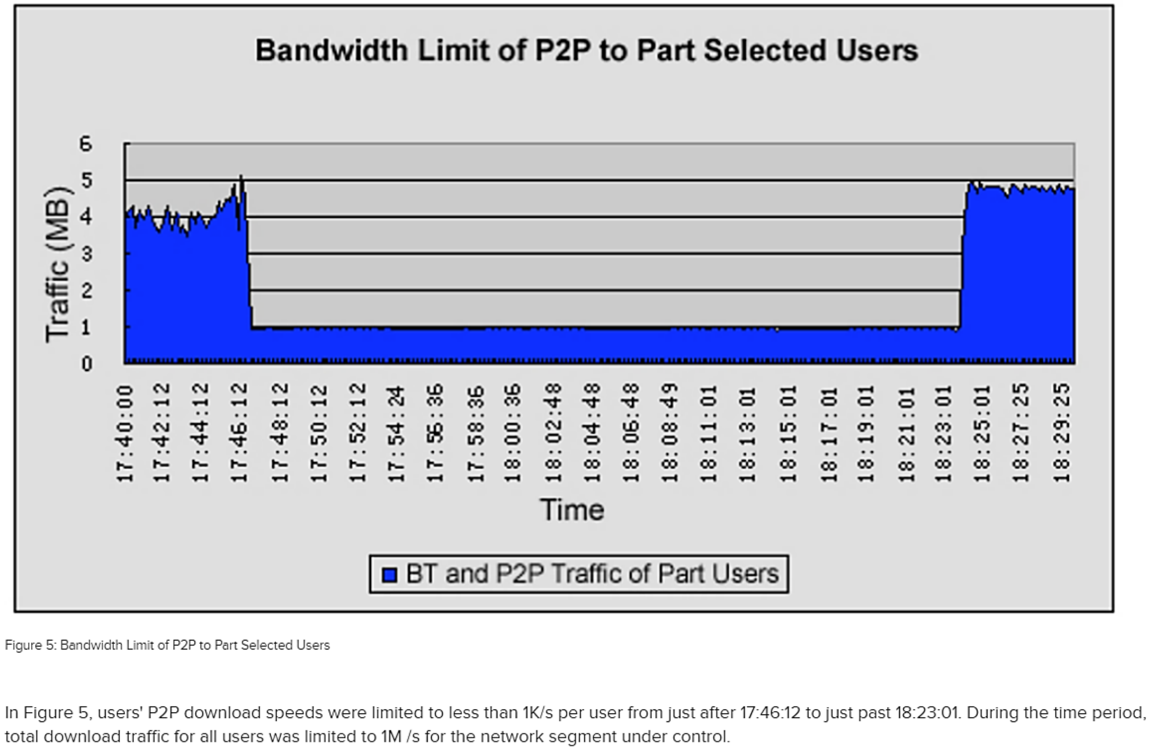


Figure 5: Bandwidth Limit of P2P to Part Selected Users

In Figure 5, users' P2P download speeds were limited to less than 1K/s per user from just after 17:46:12 to just past 18:23:01. During the time period, total download traffic for all users was limited to 1M /s for the network segment under control.

Fig 13

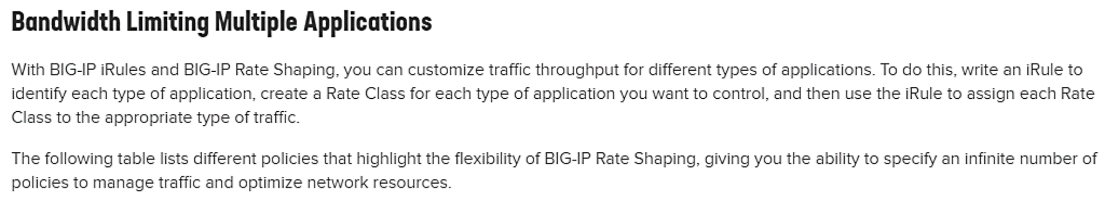
Source: https://www.f5.com/services/resources/white-papers/bandwidth-management-for-peer-to-peer-applications, Page 6, Last accessed June 19, 2020, Exhibit B

BIG-IP iRules and BIG-IP Rate Shaping can be used to customize traffic throughput for different types of applications. For this, a user can write a policy (iRule) to identify each type of application,

Exhibit 2

create a Rate Class for each type of application to be controlled, and then use the iRule to assign each Rate Class to the appropriate type of traffic. See Fig 14 and Fig 15.

Citation 14: Bandwidth Limiting



Bandwidth Limiting Multiple Applications

With BIG-IP iRules and BIG-IP Rate Shaping, you can customize traffic throughput for different types of applications. To do this, write an iRule to identify each type of application, create a Rate Class for each type of application you want to control, and then use the iRule to assign each Rate Class to the appropriate type of traffic.

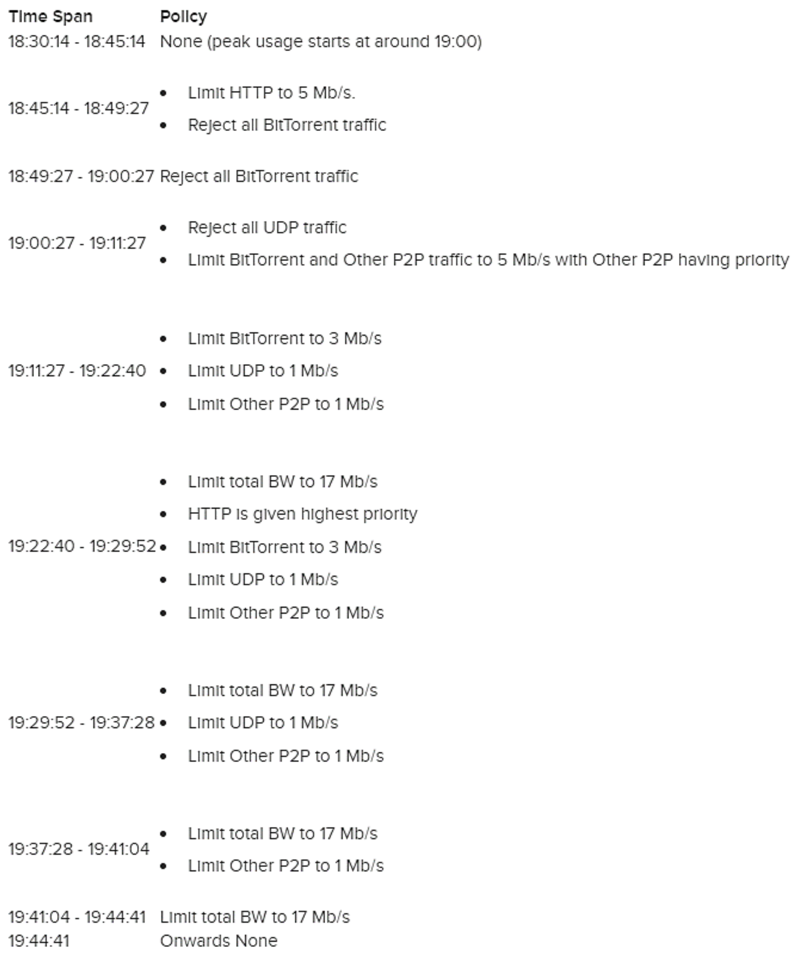
The following table lists different policies that highlight the flexibility of BIG-IP Rate Shaping, giving you the ability to specify an infinite number of policies to manage traffic and optimize network resources.

Fig 14

Source: https://www.f5.com/services/resources/white-papers/bandwidth-management-for-peer-to-peer-applications, Page 7, Last accessed June 19, 2020, Exhibit B

Exhibit 2

Citation 15: Policy



Time Span	Policy
18:30:14 - 18:45:14	None (peak usage starts at around 19:00)
18:45:14 - 18:49:27	• Limit HTTP to 5 Mb/s. • Reject all BitTorrent traffic
18:49:27 - 19:00:27	Reject all BitTorrent traffic
19:00:27 - 19:11:27	• Reject all UDP traffic • Limit BitTorrent and Other P2P traffic to 5 Mb/s with Other P2P having priority
19:11:27 - 19:22:40	• Limit BitTorrent to 3 Mb/s • Limit UDP to 1 Mb/s • Limit Other P2P to 1 Mb/s
19:22:40 - 19:29:52	• Limit total BW to 17 Mb/s • HTTP is given highest priority • Limit BitTorrent to 3 Mb/s • Limit UDP to 1 Mb/s • Limit Other P2P to 1 Mb/s
19:29:52 - 19:37:28	• Limit total BW to 17 Mb/s • Limit UDP to 1 Mb/s • Limit Other P2P to 1 Mb/s
19:37:28 - 19:41:04	• Limit total BW to 17 Mb/s • Limit Other P2P to 1 Mb/s
19:41:04 - 19:44:41	Limit total BW to 17 Mb/s
19:44:41	Onwards None

Fig 15

Exhibit 2

		Source: https://www.f5.com/services/resources/white-papers/bandwidth-management-for-peer-to-peer-applications, Page 7-8, Last accessed June 19, 2020, Exhibit B As an example, the traffic from a particular application called BitTorrent can be rejected for a certain period. During this period, the application will remain inactive and remain active for the other times, as evident from the provided bitrate and guaranteed bit rate of the BitTorrent (class). According to the patent specification, *"The AF follows the user behavior and defines the percentage of the time that the user (the UE associated with the user) is active."* [Col. 2, Line 22-24]. Therefore, an activity factor must be there in BIG-IP LTM to judge for what part (percentage) of time a particular application remains active. E.g., see Fig 16. Note: Product Testing/ Source code review is needed to strengthen the claim limitation *"wherein the activity factor defines a percentage of time that a user is active"*.

Exhibit 2

Citation 16: Bandwidth Limiting

Figure 7 illustrates the effect of four different policies used to manage four different types of traffic.

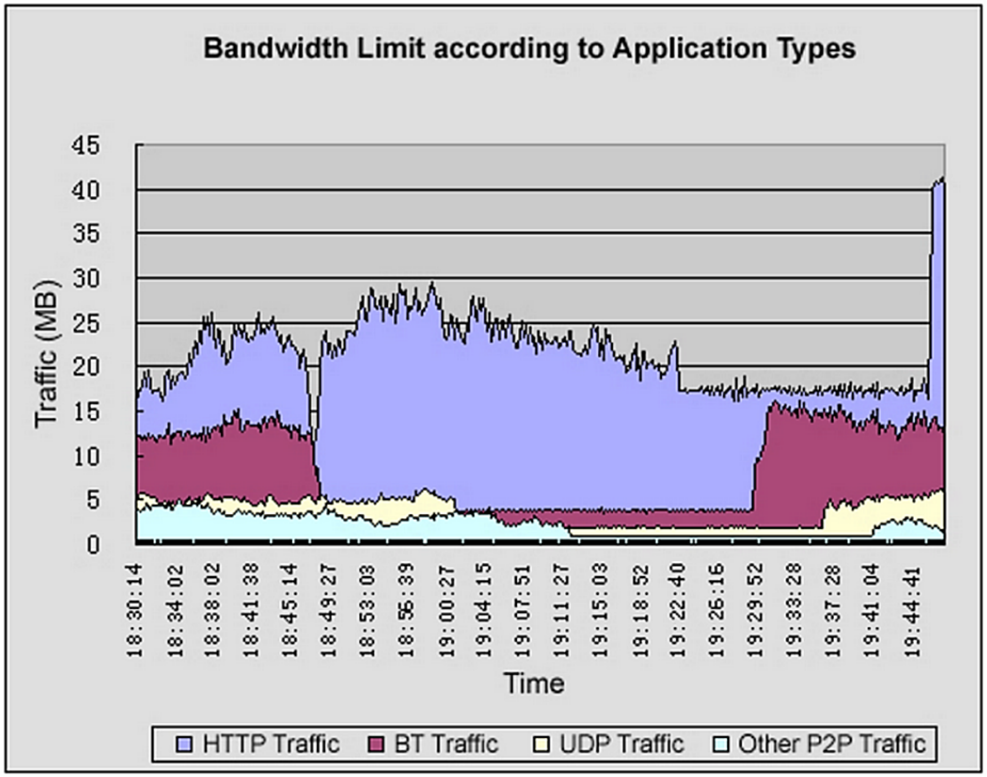


Figure 7: Effects of Bandwidth Limit Policies in Test Cases

Fig 16

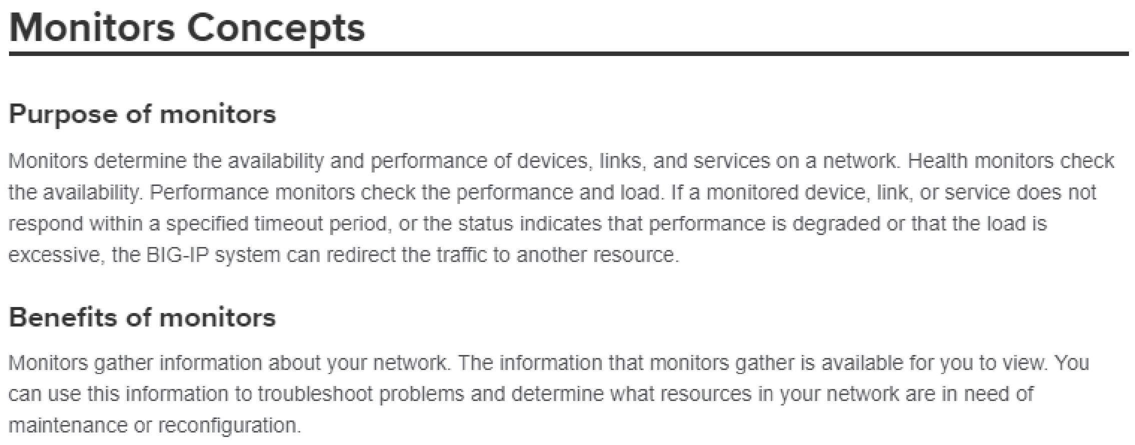
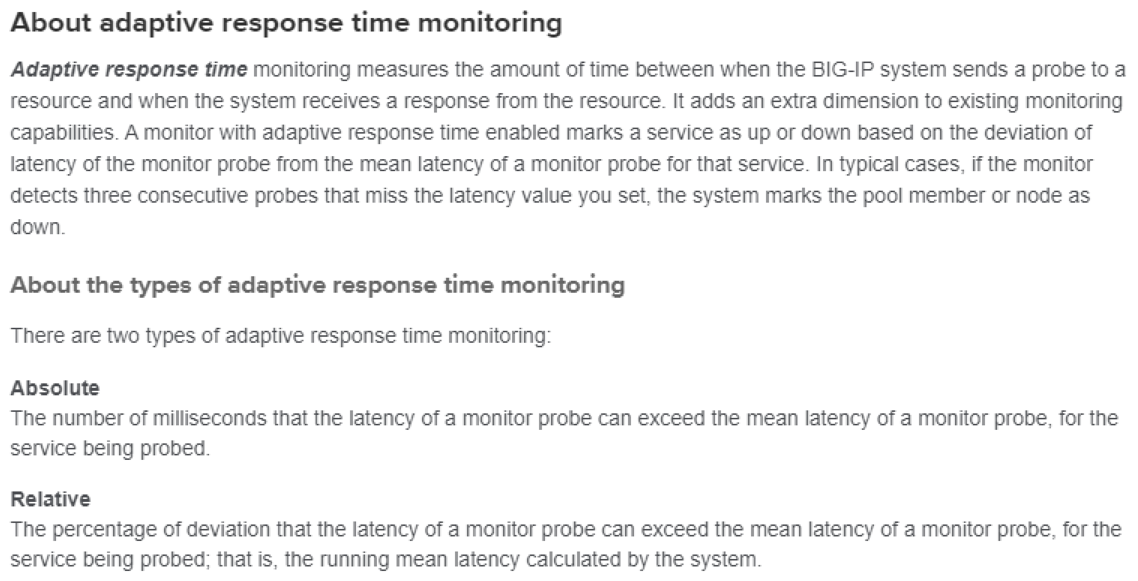
		Source: https://www.f5.com/services/resources/white-papers/bandwidth-management-for-peer-to-peer-applications, Page 8, Last accessed June 19, 2020, Exhibit B
1b. using, by the processor, the estimated activity factor to estimate at least one network-related parameter.	**An output of the AF estimation unit may be used to estimate some network-related parameter, such as the power used for existing HSDPA connections and/or a QoS factor in terms of bit rate**. The unit includes means to average the estimated AF per SPI overall HS-DSCH connections belonging to the SPI group. The SPI may be a best effort SPI, and in this case the guaranteed bit rate is a minimum guaranteed bit rate. [Col. 7, Line 33-40]	The accused product practices a method for using, by the processor, the estimated activity factor to estimate at least one network-related parameter. BIG-IP devices including the BIG-IP LTM (*i.e.,* the processor) make use of monitors to determine the availability and performance of devices, links and services on a network. Monitors are used to gather information about the network and the gathered information can be viewed by the user. See Fig 17. **Citation 17: Monitors in BIG-IP LTM**  Monitors Concepts Purpose of monitors Monitors determine the availability and performance of devices, links, and services on a network. Health monitors check the availability. Performance monitors check the performance and load. If a monitored device, link, or service does not respond within a specified timeout period, or the status indicates that performance is degraded or that the load is excessive, the BIG-IP system can redirect the traffic to another resource. Benefits of monitors Monitors gather information about your network. The information that monitors gather is available for you to view. You can use this information to troubleshoot problems and determine what resources in your network are in need of maintenance or reconfiguration. Fig 17 Source: https://techdocs.f5.com/en-us/bigip-16-0-0/big-ip-local-traffic-manager-monitors-reference/monitors-concepts.html#GUID-886B6012-8FFB-4F82-A3FE-ED263ED27CFC, Page 1, Last Accessed September 11, 2020, Exhibit C

Exhibit 2

The BIG-IP system offers adaptive response time monitoring, which measures the amount of time between when the BIG-IP system sends a probe to a resource and when the system receives a response from the resource. A monitor with adaptive response time enabled marks a service as up or down based on the deviation of latency of the monitor probe from the mean latency of a monitor probe for that service. See Fig 18.

Citation 18: Adaptive response time monitoring



About adaptive response time monitoring

Adaptive response time monitoring measures the amount of time between when the BIG-IP system sends a probe to a resource and when the system receives a response from the resource. It adds an extra dimension to existing monitoring capabilities. A monitor with adaptive response time enabled marks a service as up or down based on the deviation of latency of the monitor probe from the mean latency of a monitor probe for that service. In typical cases, if the monitor detects three consecutive probes that miss the latency value you set, the system marks the pool member or node as down.

About the types of adaptive response time monitoring

There are two types of adaptive response time monitoring:

Absolute
The number of milliseconds that the latency of a monitor probe can exceed the mean latency of a monitor probe, for the service being probed.

Relative
The percentage of deviation that the latency of a monitor probe can exceed the mean latency of a monitor probe, for the service being probed; that is, the running mean latency calculated by the system.

Fig 18

Source: https://techdocs.f5.com/en-us/bigip-16-0-0/big-ip-local-traffic-manager-monitors-reference/monitors-concepts.html#GUID-886B6012-8FFB-4F82-A3FE-ED263ED27CFC, Page 7, Last Accessed September 11, 2020, Exhibit C

Exhibit 2

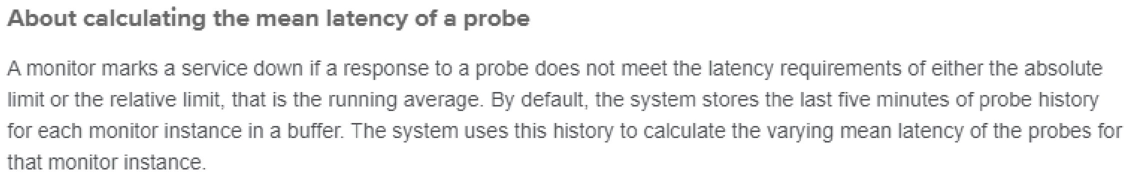
		The monitor marks a service down if a response to the probe does not meet the latency requirements of either the absolute or the relative limit. The system stores the last five minutes of probe history for each monitor instance in a buffer. The system uses this history to calculate the varying mean latency (*i.e.,* estimating a network-related parameter) of the probes for that monitor instance. See Fig 19. **Citation 19: Calculation of mean latency of a probe**  **About calculating the mean latency of a probe** A monitor marks a service down if a response to a probe does not meet the latency requirements of either the absolute limit or the relative limit, that is the running average. By default, the system stores the last five minutes of probe history for each monitor instance in a buffer. The system uses this history to calculate the varying mean latency of the probes for that monitor instance. Fig 19 Source: https://techdocs.f5.com/en-us/bigip-16-0-0/big-ip-local-traffic-manager-monitors-reference/monitors-concepts.html#GUID-886B6012-8FFB-4F82-A3FE-ED263ED27CFC, Page 7, Last Accessed September 11, 2020, Exhibit C When a web application is overwhelmed with traffic, the application may consume excessive amounts of memory and start swapping to disk. This degrades the network performance and a server may be marked "down" unnecessarily. The servers can be configured with a HTTP monitor by enabling the Adaptive setting which calculates the mean latency (*i.e.,* network-related parameter). These parameters consider the provided bit rate to the application and thus, based on the activity factor. See Fig 20.

Exhibit 2

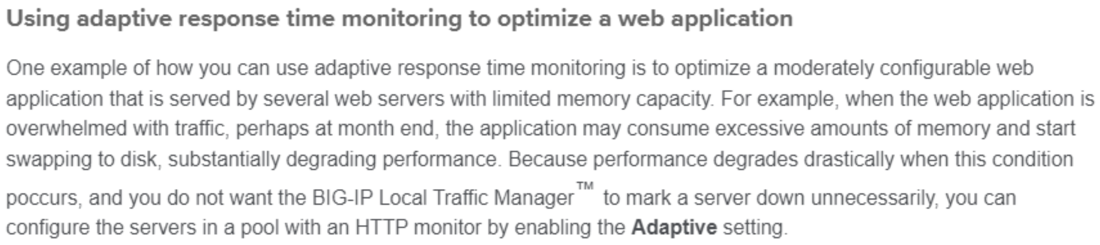
		Citation 20: Adaptive response time monitoring for web application traffic  **Using adaptive response time monitoring to optimize a web application** One example of how you can use adaptive response time monitoring is to optimize a moderately configurable web application that is served by several web servers with limited memory capacity. For example, when the web application is overwhelmed with traffic, perhaps at month end, the application may consume excessive amounts of memory and start swapping to disk, substantially degrading performance. Because performance degrades drastically when this condition poccurs, and you do not want the BIG-IP Local Traffic Manager™ to mark a server down unnecessarily, you can configure the servers in a pool with an HTTP monitor by enabling the **Adaptive** setting. Fig 20 Source: https://techdocs.f5.com/en-us/bigip-16-0-0/big-ip-local-traffic-manager-monitors-reference/monitors-concepts.html#GUID-886B6012-8FFB-4F82-A3FE-ED263ED27CFC, Page 8, Last Accessed September 11, 2020, Exhibit C

Exhibit 2

References Cited

Exhibit(s)	Description	Link
Exhibit A	About Traffic Classes	https://www.f5.com/services/resources/white-papers/bandwidth-management-for-peer-to-peer-applications
Exhibit B	Bandwidth Management for Peer-to-Peer Applications	Bandwidth Management for Peer-to-Peer Applications
Exhibit C	BIG-IP Monitors	https://techdocs.f5.com/en-us/bigip-16-0-0/big-ip-local-traffic-manager-monitors-reference/monitors-concepts.html#GUID-886B6012-8FFB-4F82-A3FE-ED263ED27CFC
Exhibit D	What Is BIG-IP?	https://devcentral.f5.com/s/articles/what-is-big-ip-24596
Exhibit E	iRule	https://clouddocs.f5.com/api/irules/
Exhibit F	Rate Shaping	https://techdocs.f5.com/kb/en-us/products/big-ip-aam/manuals/product/aam-concepts-11-5-0/4.html

Exhibit A

Manual Chapter : About Traffic Classes

Applies To:

Show Versions ⊞

About Traffic Classes

About traffic classes

The BIG-IP® system includes a feature known as traffic classes. A traffic class is a feature that you can use when implementing optimization profiles for modules such as the Application Acceleration Manager™ .

A ***traffic class*** allows you to classify traffic according to a set of criteria that you define, such as source and destination IP addresses. In creating the traffic class, you define not only classification criteria, but also a classification ID. Once you have defined the traffic class and assigned the class to a virtual server, the system associates the ***classification ID*** to each traffic flow. In this way, the system can regulate the flow of traffic based on that classification.

When attempting to match traffic flows to a traffic class, the system uses the most specific match possible.

Creating a traffic class

By creating a traffic class and assigning it to a virtual server, you can classify traffic according to a set of criteria that you specify.

1. On the Main tab, click **Local Traffic › Traffic Class**.
2. Click the **Create** button.
3. In the **Name** field, type a name for the traffic class.

   Traffic class names are case-sensitive and can contain letters, numbers, and underscores (_) only.
4. In the **Classification** field, type a text string that the system applies to data flows that match the traffic-class criteria.
   When values from a traffic flow match the criteria specified on this screen, the system tags the traffic flow with this classification value.

5. In the **Source Address** field, type an IP address for the system to match against incoming traffic.
6. For the **Source Mask** field, type a network mask for the specified source address.
7. For the **Source Port** setting, type a port number in the field or select a port name from the list.
   If you select a port name from the list, the system displays the corresponding port number in the text field.
8. In the **Destination Address** field, type an IP address for the system to match against the traffic destination.
9. For the **Destination Mask** field, type a network mask for the specified destination address.
10. For the **Destination Port** setting, type a port number in the field or select a port name from the list.
    If you select a port name from the list, the system displays the corresponding port number in the text field.
11. For the **IP Protocol** setting, type a protocol number in the field or select a protocol name from the list.
    If you select a protocol name from the list, the system displays the corresponding protocol number in the text field.
12. Click the **Finished** button.

After you define the traffic class and assign the class to a virtual server, the system associates the corresponding classification ID to traffic flows that match the specified criteria. In this way, the system can regulate the flow of traffic based on that classification.

Have a Question?
Support and Sales >

Follow Us

About F5

Corporate Information
Newsroom
Investor Relations
Careers
About AskF5

Education

Training
Certification
F5 University
Free Online Training

F5 Sites

F5.com
DevCentral
Support Portal
Partner Central
F5 Labs

Support Tasks

Read Support Policies
Create Service Request
Leave feedback [+]

Exhibit B

WHITE PAPER

Bandwidth Management for Peer-to-Peer Applications

UPDATED
DECEMBER 14, 2007

With the increasing proliferation of broadband, more and more users are using Peer-to-Peer (P2P) protocols to share very large files, including software, multi-media files, and applications. This trend has exponentially increased traffic flows across a very wide area network.

If you are coping with excessive bandwidth consumption due to P2P traffic such as BitTorrent, eMule, numerous Gnutella clients, DirectConnect, Kazaa, etc., conventional rate shaping techniques such as limiting bandwidth by TCP port number may not do the trick. A more powerful technique based on application signature identification via packet inspection may be needed.

Traditional rate shaping techniques may not be sufficient to control new breeds of applications. For example, BitTorrent is a protocol that is typically used by simple desktops to transfer user files via broadband connections. However, using BitTorrent to transfer high volumes of data puts huge pressures on the broadband operators' network. Unfortunately, prohibiting BitTorrent traffic has become routine for some broadband operators and is now a key area of contention between users and broadband operators.

This paper describes how you can use F5 BIG-IP iRules and the Rate Shaping feature of BIG-IP Local Traffic Manager (LTM) to identify different types of traffic for individualized control that can return double-digit capacity without spending a dime on additional bandwidth. Through the combination of iRules and Rate Shaping, you can:

- Ensure that critical applications are not impacted by non-priority traffic.
- Deliver optimal application performance by allocating more bandwidth for higher priority applications
- Eliminate special purpose Rate Shaping products for simplified, centralized traffic management capabilities
- Provide flexible bandwidth limits, bandwidth borrowing, and traffic queuing
- Control rate classes based on any traffic variable
- Enable application bandwidth to be shared across similar priority applications for better resource sharing
- Ensure that specific types of application traffic stay within authorized boundaries

Re-gaining Network Traffic Control

Rather than using a one-size-fits-all approach to controlling network traffic, network managers need a more application-oriented way to transmit and distribute network data. In the case of BitTorrent traffic, F5 suggests:

- **Step 1** Identifying BitTorrent traffic via packet inspection
- **Step 2** Implementing a rule to isolate BitTorrent traffic

With BIG-IP iRules and the Rate Shaping feature in the BIG-IP Local Traffic Management system, you can control the bandwidth usage of any type of traffic. Figure 1 shows how Rate Shaping can control the bandwidth usage of just BitTorrent traffic.

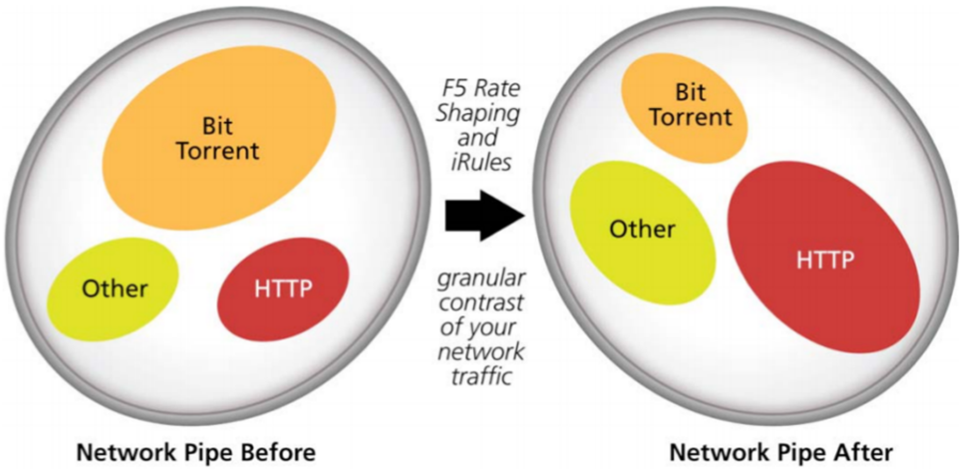


Figure 1: Controlling BitTorrent Traffic

The following sections describe each step of the process, provide a sample iRule to identify the BitTorrent application signature, and describe your options for controlling virtually any type of traffic.

Detection of BitTorrent Traffic

According to a recently published paper by AT&T Labs[1], inspection of the data packets that are transmitting between clients is a good way to detect BitTorrent traffic. The communication between BitTorrent clients starts with a handshake followed by a never-ending stream of length-prefixed messages. The header of the BitTorrent handshake message uses the following format:

< a character (1 byte)>< a string (19 byte)>

The first byte is a fixed character with value "19," and the string value is "BitTorrent protocol." Based on this common header, you can use the following signatures to identify BitTorrent traffic:

- The first byte in the TCP payload is the character 19 (0x13)
- The next 19 bytes match the string "BitTorrent protocol"

Using BIG-IP iRules to Detect BitTorrent Traffic

BIG-IP iRules is a powerful yet simple tool you can use to identify and isolate the application traffic you want to direct, filter, or persist on. BIG-IP iRules gives you the ability to customize application switching based on business needs, optimizing the handling of traffic - where and when to send it for the fastest response based on application type, category, and priority.

The following example uses an iRule to intercept traffic and pinpoint when a TCP connection has initiated BitTorrent communication and manage only that traffic without affecting any other type of traffic.

```
when CLIENT_ACCEPTED {

TCP::collect 0 0 // start data collection after client TCP handshake
connectio
is
// completed
}
when CLIENT_DATA {
append payload [TCP::payload] // assign the collected contents in
"payload"
if {[string length $payload] < 6} { // pass directly if collected contents is le
return //end Rules operation, and not carry out subsequent statements
}
TCP::release //release the collected contents and go along subsequent
work
binary scan $payload cc5 bt_size bt_protocol //analyze packet content
obtained.
if {($bt_protocol == "66 105 116 84 111") && ($bt_size == 19)} {
log "Torrent traffic from [IP::remote_addr]" // add Log if it needs to record
IP
Rate Class p2p_bt //if pattern matches, put it in Rate Class 'p2p_bt' for
processing
}
}
```

Once a TCP client is accepted, BIG-IP inspects the first packet's payload of a TCP connection and looks for a match with the BitTorrent protocol signature. Using the BIG-IP Rate Shaping feature, you can assign a Rate Class that corresponds to the policy you define to control traffic with the BitTorrent protocol signature. In this example, if the TCP payload is a BitTorrent payload type, it is assigned to the Rate Class "p2p_bt."

You can also target BitTorrent traffic for special processing, isolating it from all other traffic on the network including routing all BitTorrent traffic through a separate WAN link, limiting the amount of bandwidth devoted to BitTorrent traffic, or any combination of bandwidth control techniques described in this paper.

Once the connection is built, you can designate all the subsequent packets in the same client session as "p2p_bt," using the BIG-IP session persistence feature. BIG-IP minimizes the degradation of switching efficiencies due to packet inspection because it doesn't need to process every packet of a session beyond the first few bytes of the first payload packet.

Bandwidth Control

Some of the key bandwidth control functions used to manage user traffic by broadband operators include:

- Bandwidth limit of Peer-to-Peer (P2P) application to an individual user (IP)
- Bandwidth limit of P2P application to a group of selected users
- Bandwidth limit of specific application (BitTorrent, WWW, FTP, etc.) to an individual user (IP)
- Bandwidth limit specific applications to a selected user group
- Bandwidth limit the exit traffic according to application types

With the Rate Shaping feature, BIG-IP gives you the ability to:

- Limit bandwidth
- Control bandwidth bursting
- Limit bandwidth by direction

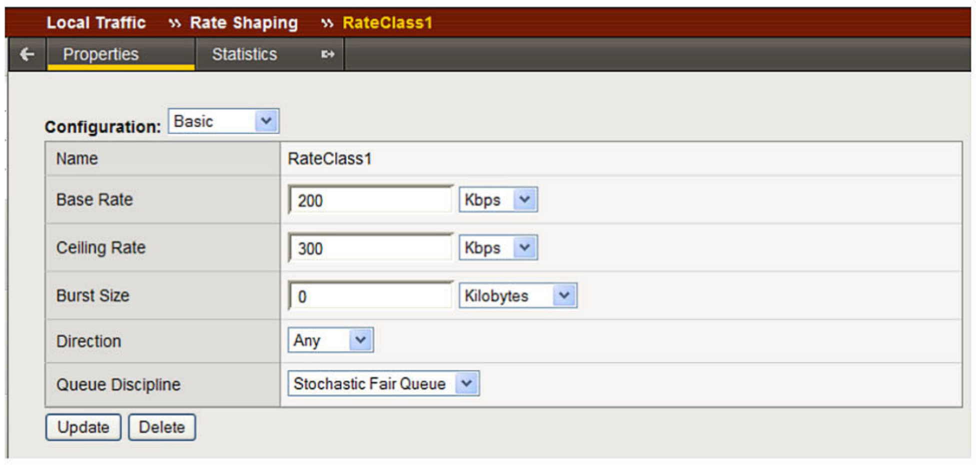
BIG-IP controls bandwidth per Rate Class, so you can control the traffic in a single Rate Class type to obey any and all rate shaping rules independent of the rules you specify for any other Rate Class. By combining bandwidth control functionality with an iRule that identifies and isolates specific types of traffic, you can control traffic in the following ways:

- Base throughput rate
- Absolute limit on the rate at which traffic is allowed to flow when bursting or borrowing
- Maximum number of bytes that traffic is allowed to burst beyond the base rate, before needing to borrow bandwidth

- Rate class from which this class can borrow bandwidth
- Method that the Rate Class uses to queue and dequeue traffic

You can also define policies in each Rate Class for traffic flowing through any single or group of virtual servers and/or pools.

The following example shows the interface and properties for a basic rate class.



Local Traffic » Rate Shaping » RateClass1

Configuration: Basic

Name	RateClass1
Base Rate	200 Kbps
Ceiling Rate	300 Kbps
Burst Size	0 Kilobytes
Direction	Any
Queue Discipline	Stochastic Fair Queue

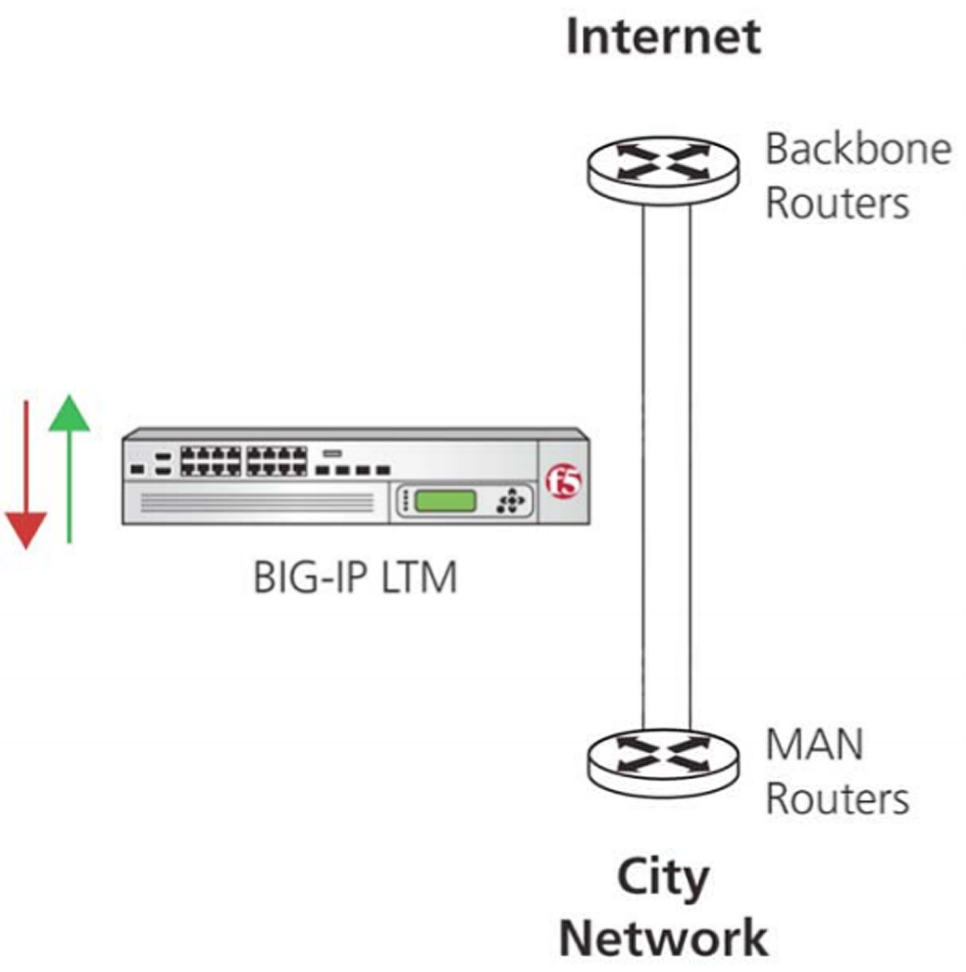
Load balancing, monitoring, and prioritizing OCSP services.

Limit Excessive, Non-critical Traffic

The typical service provider environment that facilitates P2P conversations includes a complex network of connections. These connections start from one end-user to an access network through the backbone of the core network to another access end user network, and then to the destination end users at a distant location.

A key junction point in these P2P connections, like those used with BitTorrent, is the junction point between the Metropolitan Area Network (MAN) router and the router connecting to the service provider's network backbone. From a traffic management perspective, these junction points are high-impact traffic management locations; thousands of users transverse this junction to access the service provider backbone to complete P2P file transfers. F5 traffic control at this junction point enables the network operator to manage thousands of users from one network device that is physically located at this junction point.

The customer referenced in this paper used an in-line deployment of BIG-IP LTM as a bridge between the MAN router and backbone router to manage BitTorrent traffic.



In this configuration, BIG-IP LTM bridges Giga fiber interfaces. The direct uplink port of the original MAN switch is connected through BIG-IP LTM to the backbone router, whereas the BIG-IP LTM switch acts as a bridge. A direct line connects the two routers, which are configured as a low-priority backup.

In the case of broadband operators, the ability to limit excessive non-critical traffic from gaining access to the trans-city backbone dramatically improves nationwide traffic efficiency. For a specific application with difficult-to-identify characteristics, like BitTorrent, the ability to prioritize and limit the bandwidth consumption across thousands of users from a single location is extremely valuable. And since the location of these junction points is typically in central office type facilities, using BIG-IP LTM to define traffic policies is conveniently done from a single graphical user interface.

Measuring Performance Improvements

Baselining

Implementing traffic policies starts with measuring and documenting the baseline performance of your "untuned" network. You can use any monitoring solution (MRTG, Cacti, Cricket) to baseline the performance of your network.

Bandwidth Consumption Baseline

Prior to creating policies to manage specific types of traffic, measure the traffic load passing through the BIG-IP switch against the baseline performance. You can configure a simple monitoring solution to draw curve diagrams of input/output traffic change through the system. Figure 3 shows an example of a data traffic diagram collected by a typical traffic performance monitor system.

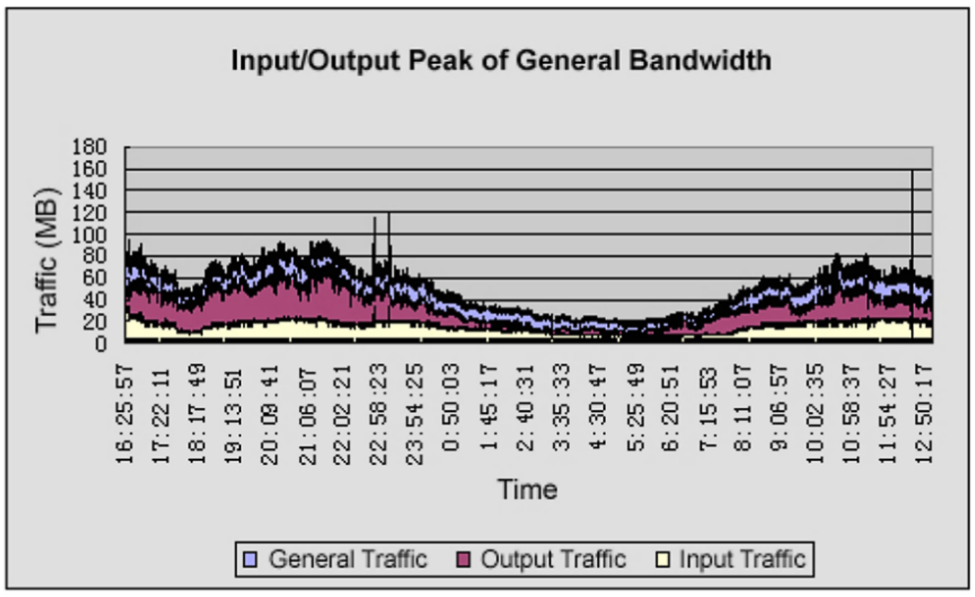


Figure 3: Input/Output Peak of General Bandwidth

Note that the traffic through this network junction is high from 8:00 to 24:00, and during this time period, traffic exceeds 60MB/s quite often.

Bandwidth Consumption of Specific Applications

Baselining traffic across the network also involves the graphing of traffic throughput by application. Monitoring software can analyze typical Internet applications to determine what type of applications merit their own Rate Class. Figure 4 shows the traffic consumption of FTP and WWW traffic that you can use to determine if these types of applications are using a disproportionate amount of bandwidth.

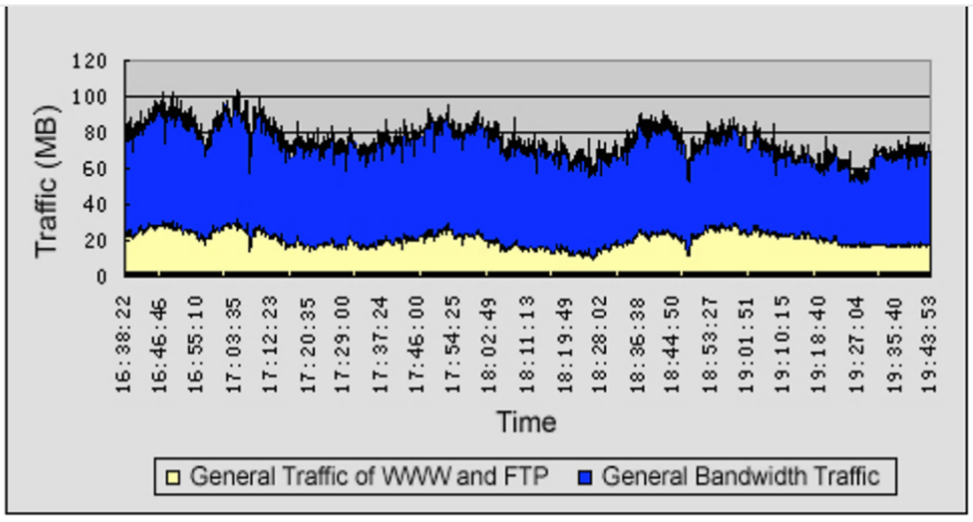


Figure 4: Bandwidth Proportion of WWW and FTP

In many situations, you may want to prioritize some types of applications as higher or lower priority. For instance, you may want to prioritize WWW and FTP traffic as lower priority traffic. However, look at the volume of WWW and FTP traffic before applying this kind of traffic policy to understand the impact of changing the traffic priority of these types of applications.

Bandwidth Control Implementation

After baselining, you can use a combination of F5 BIG-IP iRules and the Rate Shaping feature of BIG-IP LTM to improve application performance. The following sections describe policies you can create to limit bandwidth for:

- P2P traffic
- WWW applications
- Multiple types of applications

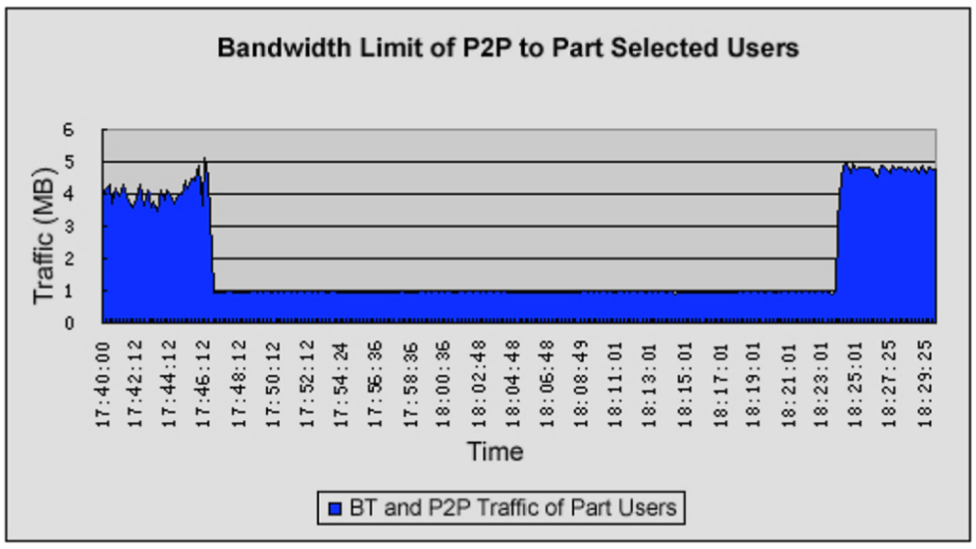
Bandwidth Limiting P2P Traffic

A common traffic management rule is limiting bandwidth for specific applications that are lower priority or consume excessive bandwidth when not controlled. In the case of service providers, bandwidth limiting P2P applications in one or more network segments and/or users is an effective way to manage this type of traffic.

With F5, you can bandwidth limit only P2P traffic and select the users to which this traffic policy applies using:

- BIG-IP iRules to analyze P2P traffic in certain IP network segments and assign certain users running P2P applications to a unique Rate Class
- BIG-IP Rate Shaping to define a policy that limits the P2P application bandwidth of the IP network segment for that Rate Class

What if you want to restrict P2P traffic to a limited amount of bandwidth only during peak use? Once configured with a Rate Class, you can watch the traffic change on the user's monitoring system, as shown in Figure 5.



In Figure 5, users' P2P download speeds were limited to less than 1K/s per user from just after 17:46:12 to just past 18:23:01. During the time period, total download traffic for all users was limited to 1M /s for the network segment under control.

Bandwidth Limit of WWW Applications

With BIG-IP LTM, you can bandwidth limit only WWW applications. To analyze the effect of this kind of traffic policy, do the following:

- Write an iRule to identify WWW applications and assign HTTP traffic to a separate Rate Class.
- Using BIG-IP Rate Shaping, create a Rate Class to limit user application bandwidth for HTTP traffic.
- Watch the traffic change on a network monitoring system.

In Figure 6, the user's HTTP traffic remains within a pre-defined range between 19:22 and 19:46.

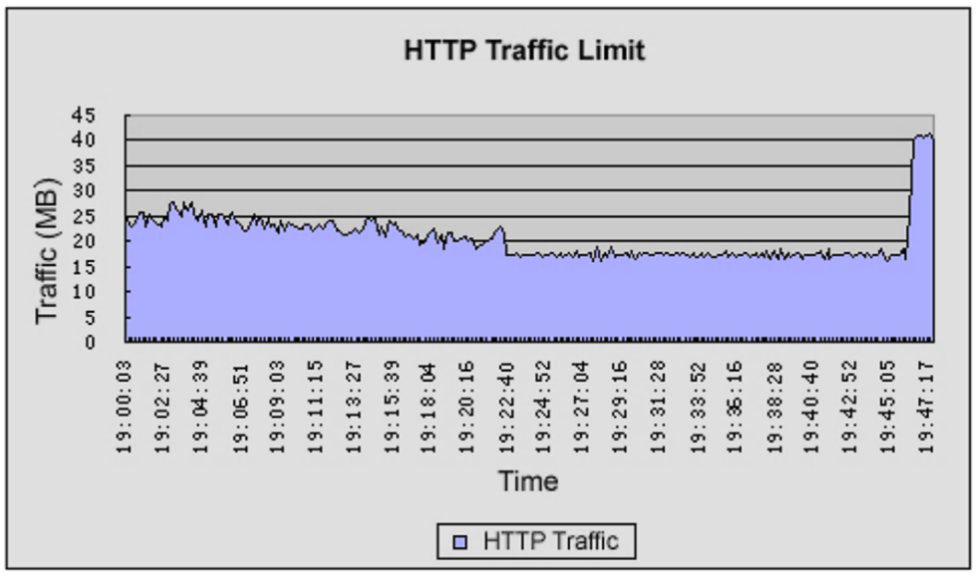


Figure 6: HTTP Traffic Limit

When HTTP is limited, pages open more slowly and HTTP application performance decreases. Once the bandwidth limit is lifted, users' HTTP traffic rises to around 40MB/s when the limit is canceled at around 19:46. HTTP traffic was limited to 18M/s while the traffic policy is in force to prevent degrading users' non-HTTP applications during the test.

Bandwidth Limiting Multiple Applications

With BIG-IP iRules and BIG-IP Rate Shaping, you can customize traffic throughput for different types of applications. To do this, write an iRule to identify each type of application, create a Rate Class for each type of application you want to control, and then use the iRule to assign each Rate Class to the appropriate type of traffic.

The following table lists different policies that highlight the flexibility of BIG-IP Rate Shaping, giving you the ability to specify an infinite number of policies to manage traffic and optimize network resources.

Time Span	Policy
18:30:14 - 18:45:14	None (peak usage starts at around 19:00)
18:45:14 - 18:49:27	• Limit HTTP to 5 Mb/s. • Reject all BitTorrent traffic
18:49:27 - 19:00:27	Reject all BitTorrent traffic

19:00:27 - 19:11:27	• Limit BitTorrent and Other P2P traffic to 5 Mb/s with Other P2P having priority
19:11:27 - 19:22:40	• Limit BitTorrent to 3 Mb/s • Limit UDP to 1 Mb/s • Limit Other P2P to 1 Mb/s
19:22:40 - 19:29:52	• Limit total BW to 17 Mb/s • HTTP is given highest priority • Limit BitTorrent to 3 Mb/s • Limit UDP to 1 Mb/s • Limit Other P2P to 1 Mb/s
19:29:52 - 19:37:28	• Limit total BW to 17 Mb/s • Limit UDP to 1 Mb/s • Limit Other P2P to 1 Mb/s
19:37:28 - 19:41:04	• Limit total BW to 17 Mb/s • Limit Other P2P to 1 Mb/s
19:41:04 - 19:44:41	Limit total BW to 17 Mb/s
19:44:41	Onwards None

Figure 7 illustrates the effect of four different policies used to manage four different types of traffic.

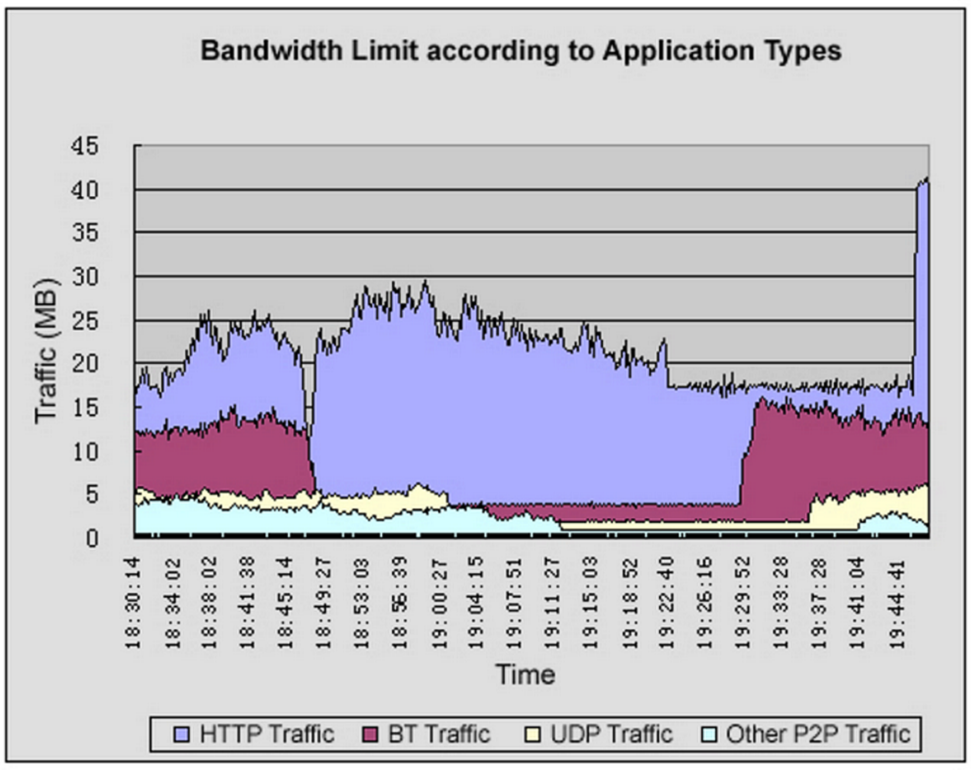


Figure 7: Effects of Bandwidth Limit Policies in Test Cases

Customer Implementation

The customer that was faced with the broadband challenge configured BIG-IP LTM to limit bandwidth by application between the MAN and the backbone. They used separate Rate Classes to manage traffic limiting:

- P2P traffic at 4MB/s
- All other HTTP traffic at 20MB/s
- eMule users at 1MB/s (detection of eMule's traffic is similar to detecting BitTorrent traffic whereas the first character of each payload packet is 0xE3, (Source: AT&T Labs - Research)

Figure 8 shows the traffic throughput before and after implementing BIG-IP LTM policies for four different types of traffic.

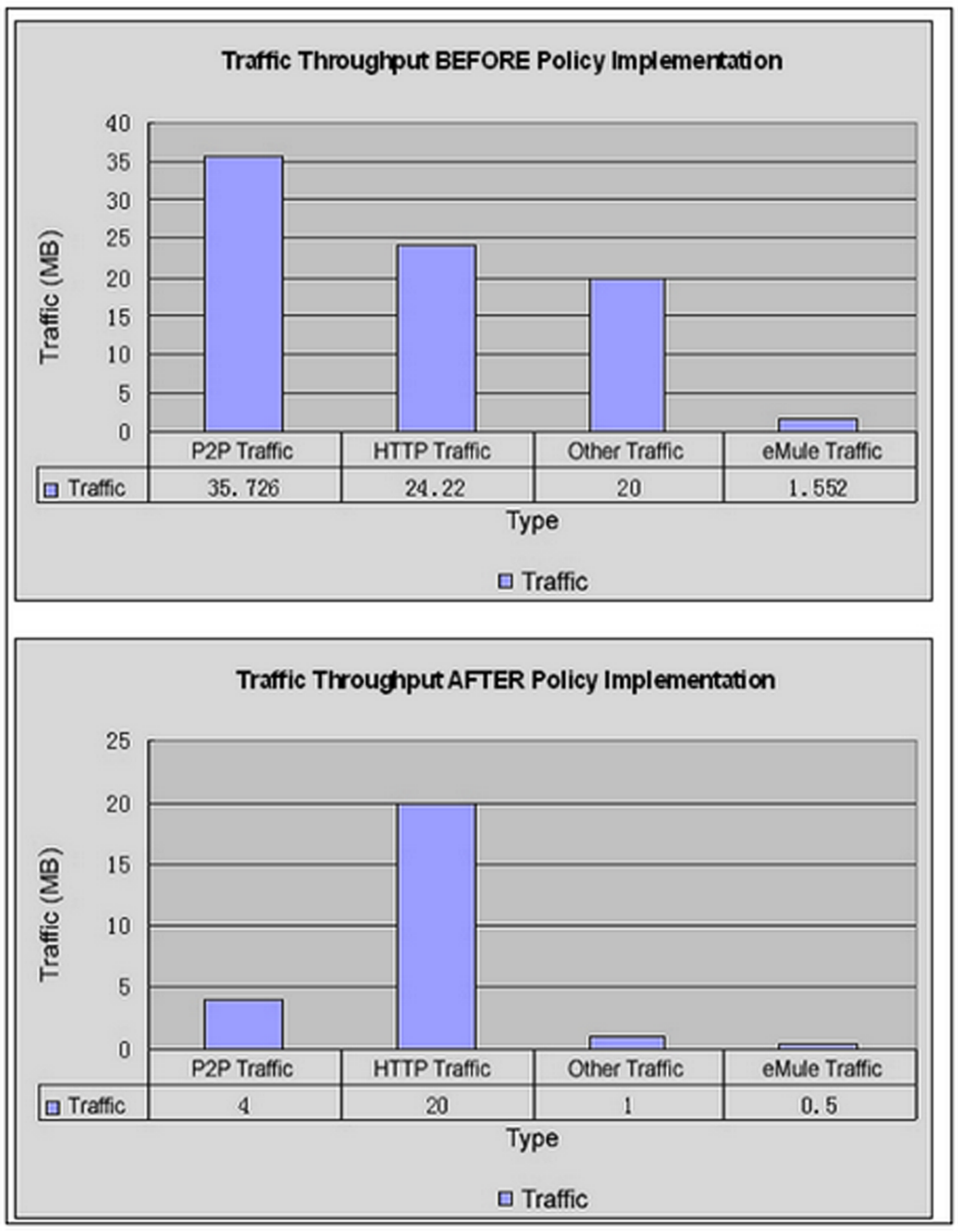


Figure 8: Effects of Traffic Management Policy Implementation

Conclusion

F5 gives service providers the ability to deftly manage the impact of otherwise uncontrollable user applications. The key capabilities that enable service providers to regain network control include:

- BIG-IP iRules: identify specific types of traffic for precise control. iRules enable BIG-IP LTM to read packet contents, identify traffic signatures within the packet, and assign all traffic with that application signature to a unique Rate Class. With iRules, you can identify the type of traffic and assign a rate class to control that type of traffic on any traffic flow variable.
- Rate Shaping: BIG-IP Rate Shaping gives you the power and flexibility to manage specific types of traffic in a variety of different ways. Because Rate Shaping is built on F5's TMOS full application proxy architecture, you control throughput in any direction (inbound, outbound). With Rate Shaping, you can create different traffic policies for each individual Rate Class to control and prioritize bandwidth usage for different types of traffic.

Although this paper focused on broadband issues, Rate Shaping capabilities also include:

- Traffic limiting, prioritization, and borrowing
- Maintaining enough bandwidth for high-priority applications and traffic
- Defining traffic and application limits
- Controlling the rate at which those resources are allowed to spike or burst
- Full support for bandwidth borrowing

- Granular traffic classification L2 through L7

1 Accurate, Scalable In-Network Identification of P2P Traffic Using Application Signatures - AT&T Labs - Research

YOUR APPS—FAST, AVAILABLE AND SECURE—IN ANY CLOUD

F5 powers applications from development through their entire life cycle so our customers can deliver differentiated, high-performing, and secure digital experiences.

HAVE A QUESTION? Support and Sales ›

FOLLOW US

48 of the Fortune 50 rely on F5

85 offices in 43 countries

20 plus years protecting apps

ABOUT F5

- Corporate Information
- Newsroom
- Investor Relations
- Careers
- Contact Information
- Communication Preferences
- Product Certifications
- Diversity & Inclusion

EDUCATION

- Training
- Professional Certification
- LearnF5
- Free Online Training

F5 SITES

- F5.com
- DevCentral
- Support Portal
- Partner Central
- F5 Labs

Policies Privacy Trademarks

Exhibit C

F5.com | Support | Community | Partners | MYF5

AskF5 Knowledge Centers Resources My Support

AskF5 Home / Knowledge Center / BIG-IP Local Traffic Manager: Monitors Reference / Monitors Concepts

Manual Chapter : Monitors Concepts

Applies To:

Show Versions ⊞

Monitors Concepts

Purpose of monitors

Monitors determine the availability and performance of devices, links, and services on a network. Health monitors check the availability. Performance monitors check the performance and load. If a monitored device, link, or service does not respond within a specified timeout period, or the status indicates that performance is degraded or that the load is excessive, the BIGIP system can redirect the traffic to another resource.

Benefits of monitors

Monitors gather information about your network. The information that monitors gather is available for you to view. You can use this information to troubleshoot problems and determine what resources in your network are in need of maintenance or reconfiguration.

About iCheck functionality for monitors

FTP, SMTP, POP3, and IMAP monitors provide inherent iCheck functionality, which reduces the load on BIG-IP systems and improves sustained monitor performance. Additionally, iCheck functionality provides smoother performance characteristics as these monitors approach full capacity.

Methods of monitoring

The BIG-IP Local Traffic Manager™, DNS, and Link Controller™ provide three methods of monitoring: simple monitoring, active monitoring, and passive monitoring.

Simple monitoring

Simple monitoring determines whether the status of a resource is up or down. Simple monitors do not monitor pool members (and therefore, individual protocols, services, or applications on a node), but only the node itself. The system contains three simple monitors, **Gateway ICMP**, **ICMP**, and **TCP_ECHO**.

Simple monitors work well when you only need to determine the up or down status of the following:

- A Local Traffic Manager node
- A BIG-IP-DNS or Link Controller server, virtual server, pool, pool member, or link

Active monitoring

Active monitoring checks the status of a pool member or node on an ongoing basis as specified. If a pool member or node does not respond within a specified timeout period, or the status of a node indicates that performance is degraded, the BIG-IP system can redirect the traffic to another pool member or node. There are many active monitors. Each active monitor checks the status of a particular protocol, service, or application. For example, one active monitor is **HTTP**. An **HTTP** monitor allows you to monitor the availability of the HTTP service on a pool, pool member, or node. A **WMI** monitor allows you to monitor the performance of a node that is running the Windows Management Instrumentation (WMI) software. Active monitors fall into two categories: Extended Content Verification (ECV) monitors for content checks, and Extended Application Verification (EAV) monitors for service checks, path checks, and application checks.

An active monitor can check for specific responses, and run with or without client traffic.

An active monitor also creates additional network traffic beyond the client request and server response and can be slow to mark a pool member as down.

Passive monitoring

Passive monitoring occurs as part of a client request. This kind of monitoring checks the health of a pool member based on a specified number of connection attempts or data request attempts that occur within a specified time period. If, after the specified number of attempts within the defined interval, the system cannot connect to the server or receive a response, or if the system receives a bad response, the system marks the pool member as down. There is only one passive monitor, called an **Inband** monitor.

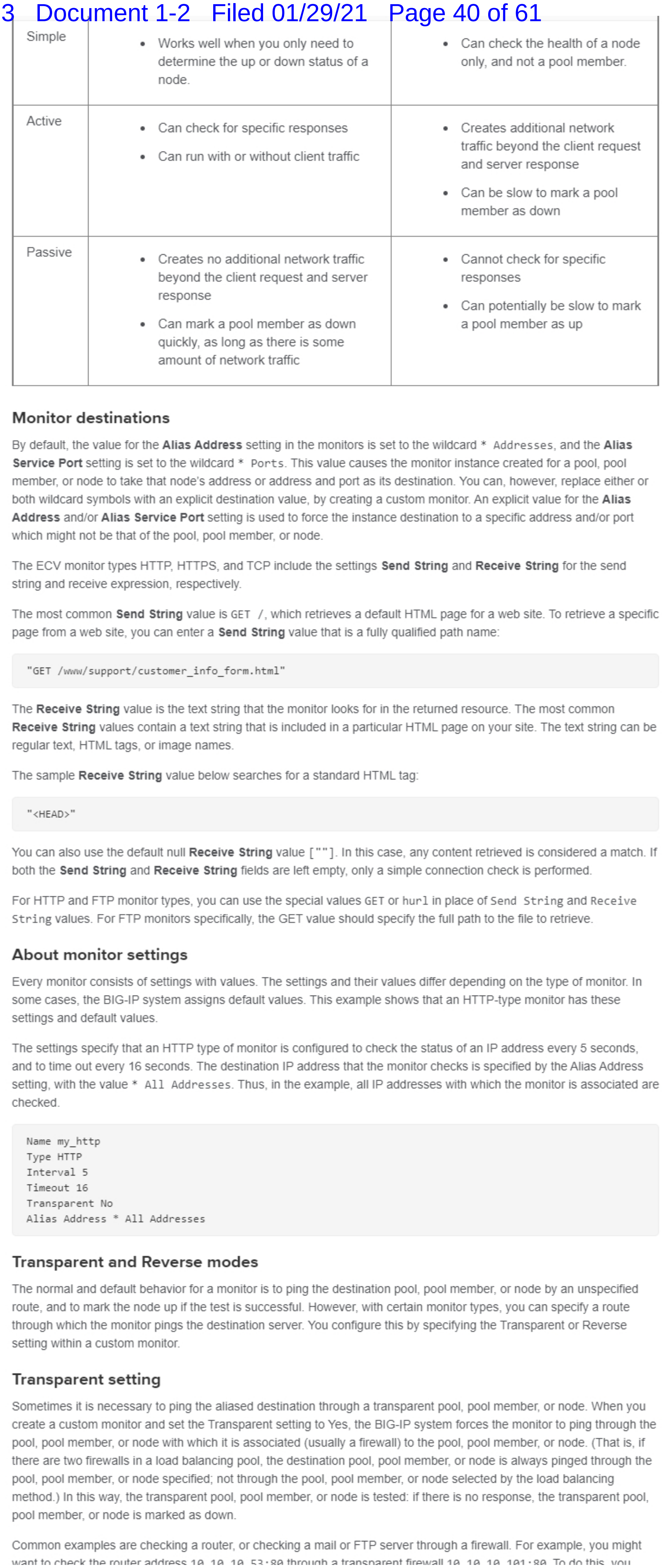
A passive monitor creates no additional network traffic beyond the client request and server response. It can mark a pool member as down quickly, as long as there is some amount of network traffic.

A passive monitor cannot check for specific responses and can potentially be slow to mark a pool member as up.

Comparison of monitoring methods

In the short description, briefly describe the purpose and intent of the information contained in this topic. This element is an F5 requirement.

Monitoring Method	Benefits	Constraints



Simple	• Works well when you only need to determine the up or down status of a node.	• Can check the health of a node only, and not a pool member.
Active	• Can check for specific responses • Can run with or without client traffic	• Creates additional network traffic beyond the client request and server response • Can be slow to mark a pool member as down
Passive	• Creates no additional network traffic beyond the client request and server response • Can mark a pool member as down quickly, as long as there is some amount of network traffic	• Cannot check for specific responses • Can potentially be slow to mark a pool member as up

Monitor destinations

By default, the value for the **Alias Address** setting in the monitors is set to the wildcard `* Addresses`, and the **Alias Service Port** setting is set to the wildcard `* Ports`. This value causes the monitor instance created for a pool, pool member, or node to take that node's address or address and port as its destination. You can, however, replace either or both wildcard symbols with an explicit destination value, by creating a custom monitor. An explicit value for the **Alias Address** and/or **Alias Service Port** setting is used to force the instance destination to a specific address and/or port which might not be that of the pool, pool member, or node.

The ECV monitor types HTTP, HTTPS, and TCP include the settings **Send String** and **Receive String** for the send string and receive expression, respectively.

The most common **Send String** value is `GET /`, which retrieves a default HTML page for a web site. To retrieve a specific page from a web site, you can enter a **Send String** value that is a fully qualified path name:

```
"GET /www/support/customer_info_form.html"
```

The **Receive String** value is the text string that the monitor looks for in the returned resource. The most common **Receive String** values contain a text string that is included in a particular HTML page on your site. The text string can be regular text, HTML tags, or image names.

The sample **Receive String** value below searches for a standard HTML tag:

```
"<HEAD>"
```

You can also use the default null **Receive String** value `[""]`. In this case, any content retrieved is considered a match. If both the **Send String** and **Receive String** fields are left empty, only a simple connection check is performed.

For HTTP and FTP monitor types, you can use the special values `GET` or `hurl` in place of `Send String` and `Receive String` values. For FTP monitors specifically, the GET value should specify the full path to the file to retrieve.

About monitor settings

Every monitor consists of settings with values. The settings and their values differ depending on the type of monitor. In some cases, the BIG-IP system assigns default values. This example shows that an HTTP-type monitor has these settings and default values.

The settings specify that an HTTP type of monitor is configured to check the status of an IP address every 5 seconds, and to time out every 16 seconds. The destination IP address that the monitor checks is specified by the Alias Address setting, with the value `* All Addresses`. Thus, in the example, all IP addresses with which the monitor is associated are checked.

```
Name my_http
Type HTTP
Interval 5
Timeout 16
Transparent No
Alias Address * All Addresses
```

Transparent and Reverse modes

The normal and default behavior for a monitor is to ping the destination pool, pool member, or node by an unspecified route, and to mark the node up if the test is successful. However, with certain monitor types, you can specify a route through which the monitor pings the destination server. You configure this by specifying the Transparent or Reverse setting within a custom monitor.

Transparent setting

Sometimes it is necessary to ping the aliased destination through a transparent pool, pool member, or node. When you create a custom monitor and set the Transparent setting to Yes, the BIG-IP system forces the monitor to ping through the pool, pool member, or node with which it is associated (usually a firewall) to the pool, pool member, or node. (That is, if there are two firewalls in a load balancing pool, the destination pool, pool member, or node is always pinged through the pool, pool member, or node specified; not through the pool, pool member, or node selected by the load balancing method.) In this way, the transparent pool, pool member, or node is tested: if there is no response, the transparent pool, pool member, or node is marked as down.

Common examples are checking a router, or checking a mail or FTP server through a firewall. For example, you might want to check the router address `10.10.10.53:80` through a transparent firewall `10.10.10.101:80`. To do this, you

To do this, you create a monitor called `http_trans` in which you specify `10.10.10.53:80` as the monitor destination address, and set the Transparent setting to Yes. Then you associate the monitor `http_trans` with the transparent pool, pool member, or node.

This causes the monitor to check the address `10.10.10 53:80` through `10.10.10.101:80`. (In other words, the BIG-IP system routes the check of `10.10.10.53:80` through `10.10.10.101:80`.) If the correct response is not received from `10.10.10.53:80`, then `10.10.10.101:80` is marked down.

Reverse setting

With the Reverse setting set to Yes, the monitor marks the pool, pool member, or node down when the test is successful. For example, if the content on your web site home page is dynamic and changes frequently, you may want to set up a reverse ECV service check that looks for the string "Error". A match for this string means that the web server was down.

Monitors that contain the Transparent or Reverse settings

This table shows the monitors that contain either the Transparent setting or both the Reverse and Transparent settings.

Monitor Type	Settings
TCP	Transparent and Reverse
HTTP	Transparent and Reverse
HTTPS	Transparent and Reverse
TCP Echo	Transparent
TCP Half Open	Transparent
ICMP	Transparent

The Manual Resume feature

By default, when a monitor detects that a resource (that is, a node or a pool member) is unavailable, the BIG-IP system marks the resource as down and routes traffic to the next appropriate resource as dictated by the active load balancing method. When the monitor next determines that the resource is available again, the BIG-IP system marks the resource as up and immediately considers the resource to be available for load balancing connection requests. While this process is appropriate for most resources, there are situations where you want to manually designate a resource as available, rather than allow the BIG-IP system to do that automatically. You can manually designate a resource as available by configuring the Manual Resume setting of the monitor.

For example, consider a monitor that you assigned to a resource to track the availability of an HTML file, *index.html*, for a web site. During the course of a business day, you decide that you need to restart the system that hosts the web site. The monitor detects the restart action and informs the BIG-IP system that the resource is now unavailable. When the system restarts, the monitor detects that the *index.html* file is available, and begins sending connection requests to the web site. However, the rest of the web site might not be ready to receive connection requests. Consequently, the BIG-IP system sends connection requests to the web site before the site can respond effectively.

To prevent this problem, you can configure the Manual Resume setting of the monitor. When you set the Manual Resume setting to Yes, you ensure that the BIG-IP system considers the resource to be unavailable until you manually enable that resource.

Resumption of connections

If you have a resource (such as a pool member or node) that a monitor marked as down, and the resource has subsequently become available again, you must manually re-enable that resource if the monitor's **Manual Resume** setting is set to Yes. Manually re-enabling the resource allows the BIG-IP system to resume sending connections to that resource.

The procedure for manually re-enabling a resource varies depending on whether the resource is a pool, a pool member, or a node.

The Time Until Up feature

By default, the BIG-IP system marks a pool member or node as up immediately upon receipt of the first correct response to a ping command.

The Time Until Up feature provides a way to adjust the default behavior. This feature allows the system to delay the marking of a pool member or node as up for some number of seconds after receipt of the first correct response. The purpose of this feature is to ensure that the monitor marks the pool member or node as up only after the pool member or node has consistently responded correctly to the BIG-IP system during the defined time period. With this feature, you ensure that a pool member or node that is available only momentarily, after sending one correct response, is not marked as up.

A Time Until Up value of `0` causes the default behavior. When the Time Until Up value is a non-0 value, the BIG-IP system marks a pool member or node as up only when all pool member or node responses during the Time Until Up period are correct.

About health and performance monitors

BIG-IP systems use two categories of monitors: health monitors and performance monitors. You can associate monitors with the following resources:

- In Local Traffic Manager: nodes, pools, and pool members
- In DNS: links, servers, virtual servers, pools, and pool members
- In Link Controller: links, pools, and pool members

Category	Description

Health	Checks resources to determine if they are up and functioning for a given service.
Performance	Gathers information about resources that the system uses to dynamically load balance traffic.

When a virtual server that is being monitored by a health monitor does not respond to a probe from the BIG-IP system within a specified timeout period, the system marks the virtual server down and no longer load balances traffic to that virtual server. When the health monitor determines that the virtual server is once again responsive, the system again begins to load balance traffic to that virtual server. To illustrate, a Gateway Internet Control Message Protocol (ICMP) monitor pings a virtual server. If the monitor does not receive a response from the virtual server, the BIG-IP system marks that virtual server down. When the ping is successful, the system marks the virtual server up.

When a server that is being monitored by a performance monitor displays a degradation in performance, the BIG-IP system redirects traffic to other resources until the performance of the server returns to normal. To illustrate, an SNMP DCA monitor checks the current CPU, memory, and disk usage of a server that is running an SNMP data collection agent, and then dynamically load balances traffic based on the performance of the server.

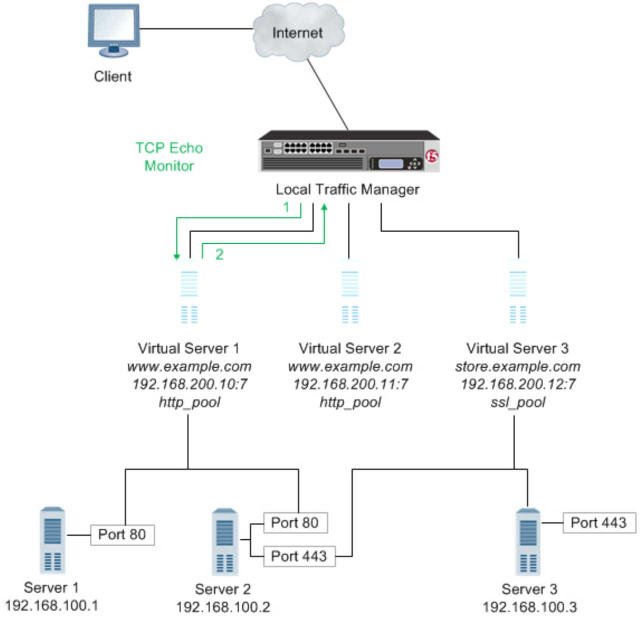
About address check monitors

An ***address check monitor*** provides a simple verification of an address on a network. This type of monitor sends a request to a virtual server. When a response is received, the test is successful.

When an address check monitor is associated with a node, it determines the availability of all services associated with that node's IP address. If the monitor is unsuccessful in determining that a node is available, the monitor marks the node and all pool members at that IP address as **Offline**.

The following illustration depicts a Local Traffic Manager using a **TCP Echo** monitor to verify an IP address for a virtual server.

Local Traffic Manager using a TCP Echo monitor



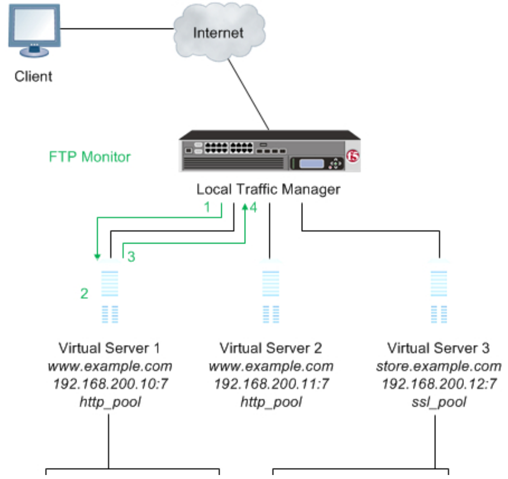
1. Local Traffic Manager sends a TCP echo request to a virtual server.
2. A TCP echo response is received.

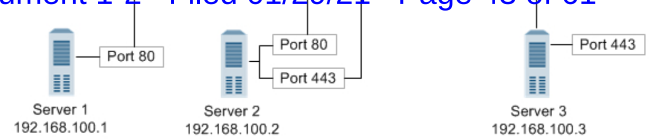
About application check monitors

An ***application check monitor*** interacts with servers by sending multiple commands and processing multiple responses.

An FTP monitor, for example, connects to a server, logs in by using a user ID and password, navigates to a specific directory, and then downloads a specific file to the /var/tmp directory. If the file is retrieved, the check is successful.

An application check monitor



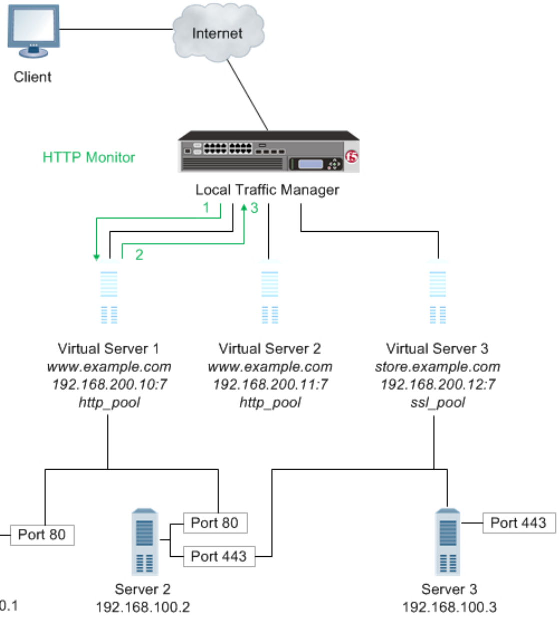


1. Local Traffic Manager opens a TCP connection to an IP address and port, and logs in to the server.
2. A specified directory is located and a specific file is requested.
3. The server sends the file to Local Traffic Manager.
4. Local Traffic Manager receives the file and closes the TCP connection.

About content check monitors

A ***content check monitor*** determines whether a service is available and whether the server is serving the appropriate content. This type of monitor opens a connection to an IP address and port, and then issues a command to the server. The response is compared to the monitor's receive rule. When a portion of the server's response matches the receive rule, the test is successful.

A content check monitor



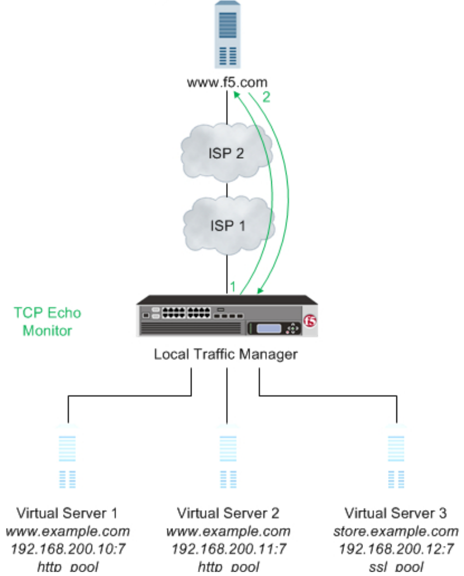
1. Local Traffic Manager opens a TCP connection to an IP address and port, and issues a command to the server.
2. The server sends a response.
3. Local Traffic Manager compares the response to the monitor's receive rule and closes the connection

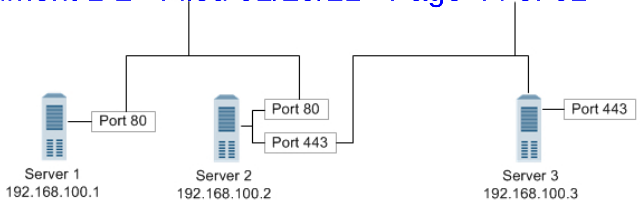
About path check monitors

A ***path check monitor*** determines whether traffic can flow through a device to an endpoint. A path check monitor is successful when network paths through firewalls or routers are available.

The following illustration depicts Local Traffic Manager (LTM) using a **TCP Echo** monitor to verify a path to a virtual server.

Local Traffic Manager using a TCP Echo monitor





1. With the **TCP Echo** monitor **Transparent** option set to **Yes**, Local Traffic Manager sends a TCP Echo request to a virtual server.
2. A TCP Echo response is received.

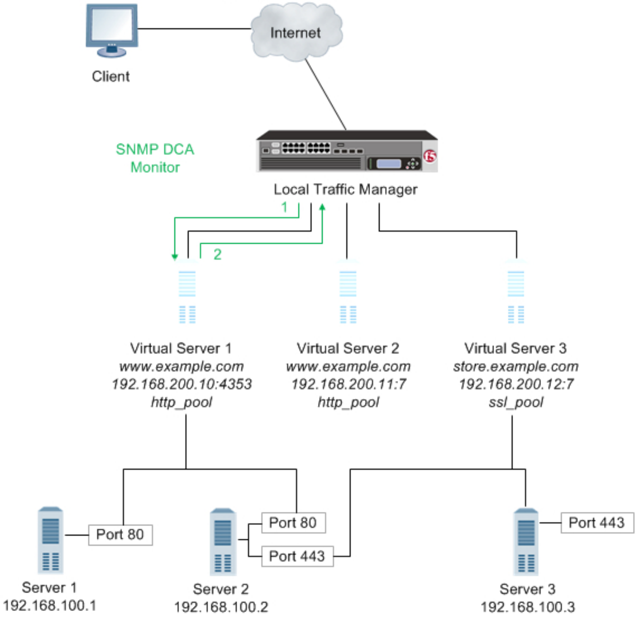
About performance check monitors

A ***performance check monitor*** interacts with servers to determine the server load, and to acquire information about the condition of virtual servers.

An SNMP DCA monitor, for example, checks the current CPU, memory, and disk usage of a pool, pool member, or node that is running an SNMP data collection agent, and then dynamically load balances traffic accordingly.

If you configure a performance monitor, such as the SNMP DCA or WMI monitor type, you should also configure a health monitor. Configuring a health monitor ensures that Local Traffic Manager reports accurate node availability status.

A performance check monitor



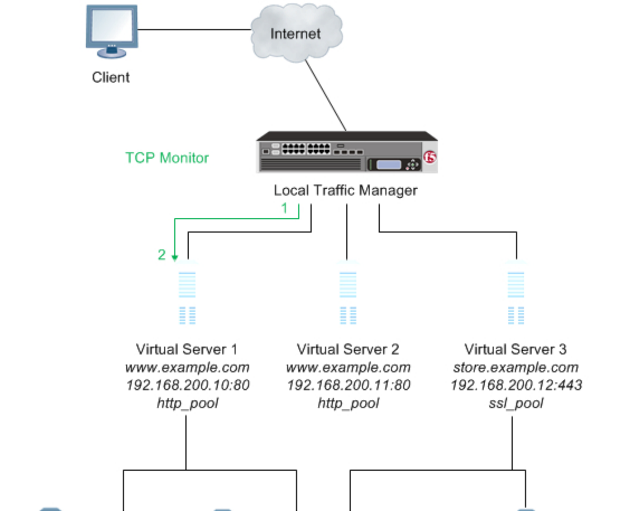
1. Local Traffic Manager connects with a server to acquire data.
2. The server sends the data to Local Traffic Manager for evaluation and determination of load balancing.

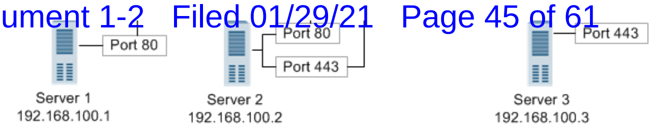
About service check monitors

A ***service check monitor*** determines whether a service is available. This type of monitor opens a connection to an IP address and port, and then closes the connection. When the TCP connection is established, the test is successful.

When a service check monitor is associated with pool members, it determines the availability of a service. If the monitor is unsuccessful in determining that a pool member is available, the monitor marks the pool member as **Offline** and no requests are sent to that pool member.

A service check monitor





1. Local Traffic Manager opens a TCP connection to an IP address and port.
2. The TCP connection is closed.

About resources and monitor queries

Network resources often perform different functions at the same time. Therefore, it is likely that multiple monitors are checking the availability of a single resource in different ways.

```
Example:
A BIG-IP system may monitor a single resource to verify that the connection to the resource
is available, that a specific HTML page on the resource can be reached, and that a database
query returns an expected result.
```

About the Virtual Location monitor

The **Virtual Location** monitor optimizes the way that the BIG-IP system manages connections to pool members by assigning priority groups to local and remote pool members.

The monitor determines whether a pool member is local (residing in the same data center as the BIG-IP system) or remote (residing in a different data center). If a pool member is local, the monitor sets the priority group of the pool member to a higher priority. If a pool member is remote, the monitor sets the priority group of the pool member to a lower priority.

You must configure Priority Group Activation to specify the minimum number of available members, before the BIG-IP system begins directing traffic to members in a lower priority group.

About adaptive response time monitoring

Adaptive response time monitoring measures the amount of time between when the BIG-IP system sends a probe to a resource and when the system receives a response from the resource. It adds an extra dimension to existing monitoring capabilities. A monitor with adaptive response time enabled marks a service as up or down based on the deviation of latency of the monitor probe from the mean latency of a monitor probe for that service. In typical cases, if the monitor detects three consecutive probes that miss the latency value you set, the system marks the pool member or node as down.

About the types of adaptive response time monitoring

There are two types of adaptive response time monitoring:

Absolute
The number of milliseconds that the latency of a monitor probe can exceed the mean latency of a monitor probe, for the service being probed.

Relative
The percentage of deviation that the latency of a monitor probe can exceed the mean latency of a monitor probe, for the service being probed; that is, the running mean latency calculated by the system.

You can enable the adaptive response time monitoring feature on these specific monitors:

- DNS
- Gateway ICMP
- HTTP
- HTTPS
- ICMP
- TCP
- UDP

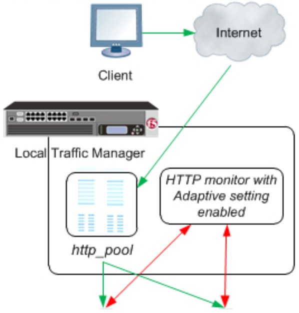
About calculating the mean latency of a probe

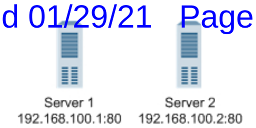
A monitor marks a service down if a response to a probe does not meet the latency requirements of either the absolute limit or the relative limit, that is the running average. By default, the system stores the last five minutes of probe history for each monitor instance in a buffer. The system uses this history to calculate the varying mean latency of the probes for that monitor instance.

How does adaptive response time monitoring work?

This example shows a BIG-IP Local Traffic Manager™ system configured to handle HTTP traffic using a pool and an HTTP monitor with adaptive response time monitoring enabled (through the **Adaptive** setting).

HTTP adaptive response time monitoring





1. A client makes an HTTP request. The HTTP request is represented by the green arrows.
2. The request is routed to an HTTP pool on BIG-IP Local Traffic Manager (LTM).
3. The LTM routes the request to one of two servers in the pool.
4. The HTTP monitor assigned to the pool determines whether the servers are up or down based on the probe latency of each server. The probe is represented by the red arrows.

Using adaptive response time monitoring to optimize a web application

One example of how you can use adaptive response time monitoring is to optimize a moderately configurable web application that is served by several web servers with limited memory capacity. For example, when the web application is overwhelmed with traffic, perhaps at month end, the application may consume excessive amounts of memory and start swapping to disk, substantially degrading performance. Because performance degrades drastically when this condition poccurs, and you do not want the BIG-IP Local Traffic Manager™ to mark a server down unnecessarily, you can configure the servers in a pool with an HTTP monitor by enabling the **Adaptive** setting.

Using adaptive response time monitoring to mitigate probe attacks

You can use adaptive response time monitoring to mitigate probe attacks. For example, consider the scenario where a popular web application for a financial company receives a a huge number of brute-force logon attempts that cause the web servers to become unresponsive. As the administrator, you can place the web servers in a pool configured for priority-based load balancing and assign an HTTP monitor with the **Adaptive** setting `Enabled`. When probe latency spikes, the monitor marks the primary servers in the pool down. When all the primary servers are marked down, the system sends requests to a secondary set of servers in the pool that presents a page that does not accept logon attempts.

Overview of monitor implementation

You implement monitors by using either the BIG-IP Configuration utility or a command line utility. The task of implementing a monitor varies depending on whether you are using a preconfigured monitor or creating a custom monitor. A ***preconfigured monitor*** is an existing monitor that BIG-IP system provides for you, with its settings already configured. A ***custom monitor*** is a monitor that you create based on one of the allowed monitor types.

If you want to implement a preconfigured monitor, you need only associate the monitor with a pool, pool member, or node, and then configure the virtual server to reference the relevant pool. If you want to implement a custom monitor, you must first create the custom monitor. Then you can associate the custom monitor with a pool, pool member, or node, and configure the virtual server to reference the pool.

Preconfigured monitors

For a subset of monitor types, the BIG-IP system includes a set of preconfigured monitors. You cannot modify preconfigured monitor settings, as they are intended to be used as is. The purpose of a preconfigured monitor is to eliminate the need for you to explicitly create a monitor. You use a preconfigured monitor when the values of the settings meet your needs as is.

Preconfigured monitors include the following entries.

- `gateway_icmp`
- `http`
- `http_head_f5`
- `https`
- `https_443`
- `https_head_f5`
- `icmp`
- `inband`
- `real_server`
- `snmp_dca`

An example of a preconfigured monitor is the `http` monitor. The example shows the `http` monitor, with values configured for its **Interval**, **Timeout**, and **Alias Address** settings. Note that the Interval value is `5`, the Timeout value is `16`, the Transparent value is `No`, and the Alias Address value is `* All Addresses`.

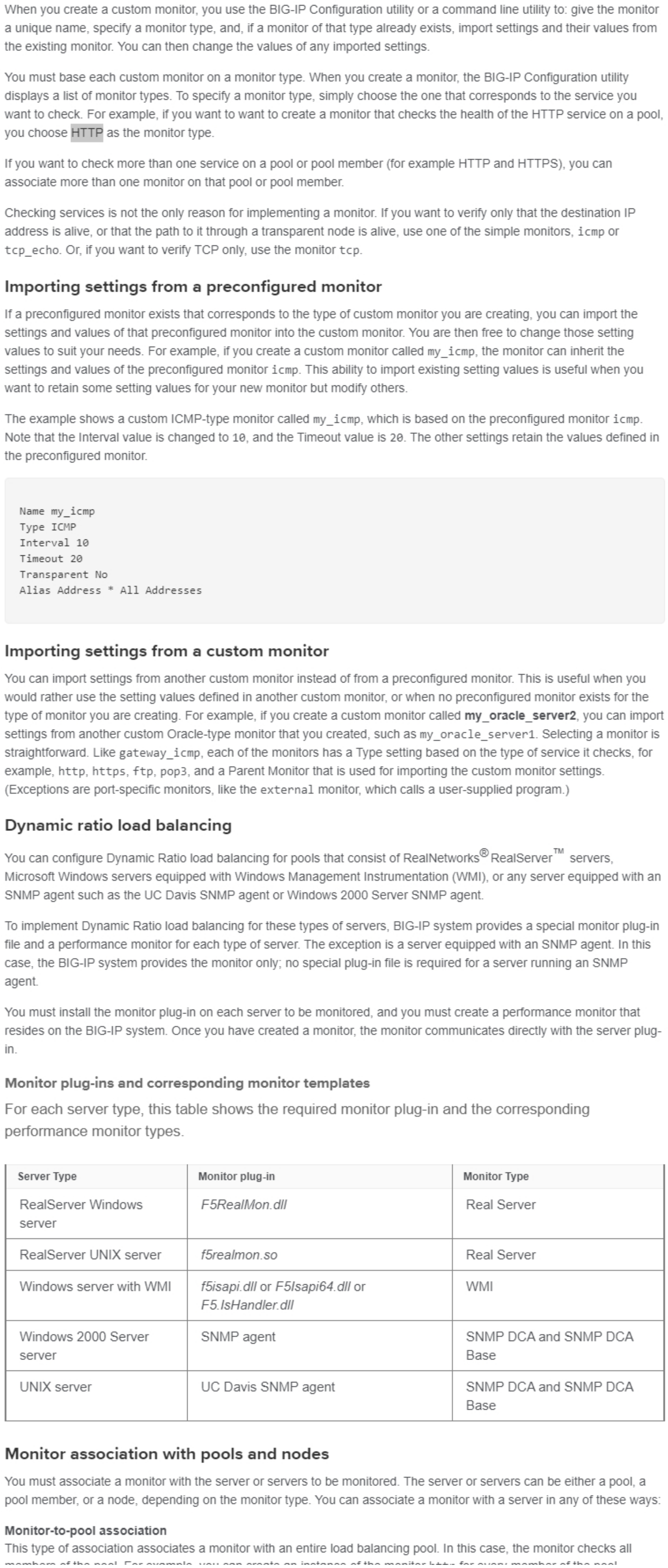
If the Interval, Timeout, Transparent, and Alias Address values meet your needs, you simply assign the `http` preconfigured monitor directly to a server, virtual server, pool, pool member, or link. In this case, you do not need to use the Monitors screens, unless you simply want to view the values of the preconfigured monitor settings.

```
Name http
Type HTTP
Interval 5
Timeout 16
Transparent No
Alias Address * All Addresses
```

All preconfigured monitors reside in partition `Common`.

Custom monitors

You create a custom monitor when the values defined in a preconfigured monitor do not meet your needs, or no preconfigured monitor exists for the type of monitor you are creating.



When you create a custom monitor, you use the BIG-IP Configuration utility or a command line utility to: give the monitor a unique name, specify a monitor type, and, if a monitor of that type already exists, import settings and their values from the existing monitor. You can then change the values of any imported settings.

You must base each custom monitor on a monitor type. When you create a monitor, the BIG-IP Configuration utility displays a list of monitor types. To specify a monitor type, simply choose the one that corresponds to the service you want to check. For example, if you want to want to create a monitor that checks the health of the HTTP service on a pool, you choose HTTP as the monitor type.

If you want to check more than one service on a pool or pool member (for example HTTP and HTTPS), you can associate more than one monitor on that pool or pool member.

Checking services is not the only reason for implementing a monitor. If you want to verify only that the destination IP address is alive, or that the path to it through a transparent node is alive, use one of the simple monitors, `icmp` or `tcp_echo`. Or, if you want to verify TCP only, use the monitor `tcp`.

Importing settings from a preconfigured monitor

If a preconfigured monitor exists that corresponds to the type of custom monitor you are creating, you can import the settings and values of that preconfigured monitor into the custom monitor. You are then free to change those setting values to suit your needs. For example, if you create a custom monitor called `my_icmp`, the monitor can inherit the settings and values of the preconfigured monitor `icmp`. This ability to import existing setting values is useful when you want to retain some setting values for your new monitor but modify others.

The example shows a custom ICMP-type monitor called `my_icmp`, which is based on the preconfigured monitor `icmp`. Note that the Interval value is changed to `10`, and the Timeout value is `20`. The other settings retain the values defined in the preconfigured monitor.

```
Name my_icmp
Type ICMP
Interval 10
Timeout 20
Transparent No
Alias Address * All Addresses
```

Importing settings from a custom monitor

You can import settings from another custom monitor instead of from a preconfigured monitor. This is useful when you would rather use the setting values defined in another custom monitor, or when no preconfigured monitor exists for the type of monitor you are creating. For example, if you create a custom monitor called **my_oracle_server2**, you can import settings from another custom Oracle-type monitor that you created, such as `my_oracle_server1`. Selecting a monitor is straightforward. Like `gateway_icmp`, each of the monitors has a Type setting based on the type of service it checks, for example, `http`, `https`, `ftp`, `pop3`, and a Parent Monitor that is used for importing the custom monitor settings. (Exceptions are port-specific monitors, like the `external` monitor, which calls a user-supplied program.)

Dynamic ratio load balancing

You can configure Dynamic Ratio load balancing for pools that consist of RealNetworks® RealServer™ servers, Microsoft Windows servers equipped with Windows Management Instrumentation (WMI), or any server equipped with an SNMP agent such as the UC Davis SNMP agent or Windows 2000 Server SNMP agent.

To implement Dynamic Ratio load balancing for these types of servers, BIG-IP system provides a special monitor plug-in file and a performance monitor for each type of server. The exception is a server equipped with an SNMP agent. In this case, the BIG-IP system provides the monitor only; no special plug-in file is required for a server running an SNMP agent.

You must install the monitor plug-in on each server to be monitored, and you must create a performance monitor that resides on the BIG-IP system. Once you have created a monitor, the monitor communicates directly with the server plug-in.

Monitor plug-ins and corresponding monitor templates

For each server type, this table shows the required monitor plug-in and the corresponding performance monitor types.

Server Type	Monitor plug-in	Monitor Type
RealServer Windows server	*F5RealMon.dll*	Real Server
RealServer UNIX server	*f5realmon.so*	Real Server
Windows server with WMI	*f5isapi.dll* or *F5Isapi64.dll* or *F5.IsHandler.dll*	WMI
Windows 2000 Server server	SNMP agent	SNMP DCA and SNMP DCA Base
UNIX server	UC Davis SNMP agent	SNMP DCA and SNMP DCA Base

Monitor association with pools and nodes

You must associate a monitor with the server or servers to be monitored. The server or servers can be either a pool, a pool member, or a node, depending on the monitor type. You can associate a monitor with a server in any of these ways:

Monitor-to-pool association
This type of association associates a monitor with an entire load balancing pool. In this case, the monitor checks all members of the pool. For example, you can create an instance of the monitor `http` for every member of the pool

members of that pool. For example, you can create an instance of the monitor http for every member of the pool my_pool, thus ensuring that all members of that pool are checked.

Monitor-to-pool member association
This type of association associates a monitor with an individual pool member, that is, an IP address and service. In this case, the monitor checks only that pool member and not any other members of the pool. For example, you can create an instance of the monitor `http` for pool member `10.10.10.10:80` of `my_pool`.

A monitor associated with an individual pool member supersedes a monitor associated with that pool member's parent pool.

Monitor-to-node association
This type of association associates a monitor with a specific node. In this case, the monitor checks only the node itself, and not any services running on that node. For example, you can create an instance of the monitor `icmp` for node `10.10.10.10`. In this case, the monitor checks the specific node only, and not any services running on that node. You can designate a monitor as the default monitor that you want the BIG-IP system to associate with one or more nodes. In this case, any node to which you have not specifically assigned a monitor inherits the default monitor.

Some monitor types are designed for association with nodes only, and not pools or pool members. Other monitor types are intended for association with pools and pool members only, and not nodes. Finally, in some instances, some monitor types associated with a node are not mutually exclusive of pools or pool members, and must function in combination in some scenarios.

Node-only monitors specify a destination address in the format of an IP address with no service port (for example, `10.10.10.2`). Conversely, monitors that you can associate with nodes, pools, and pool members specify a destination address in the format of an IP address and service port (for example, `10.10.10.2:80`). Therefore, when you use the BIG-IP Configuration utility to associate a monitor with a pool, pool member, or node, the utility displays only those pre-configured monitors that are designed for association with that server.

For example, you cannot associate the monitor `icmp` with a pool or its members, since the `icmp` monitor is designed to check the status of a node itself and not any service running on that node.

Monitor instances

When you associate a monitor with a server, the BIG-IP system automatically creates an ***instance*** of that monitor for that server. A monitor association thus creates an instance of a monitor for each server that you specify. This means that you can have multiple instances of the same monitor running on your servers.

Because instances of monitors are not partitioned objects, a user can enable or disable an instance of a monitor without having permission to manage the associated pool or pool member.

For example, a user with the Manager role, who can access partition `AppA` only, can enable or disable monitor instances for a pool that resides in partition `Common`. However, that user cannot perform operations on the pool or pool members that are associated with the monitor. Although this is correct functionality, the user might not expect this behavior. You can prevent this unexpected behavior by ensuring that all pools and pool members associated with monitor instances reside in the same partition.

Have a Question? | Support and Sales >

Follow Us

About F5

Corporate Information
Newsroom
Investor Relations
Careers
About AskF5

Education

Training
Certification
F5 University
Free Online Training

F5 Sites

F5.com
DevCentral
Support Portal
Partner Central
F5 Labs

Support Tasks

Read Support Policies
Create Service Request
Leave feedback [+]

Exhibit D

(/s/) F5.COM (https://f5.com/) | DEVCENTRAL (/s/) | SUPPORT (https://support.f5.com/) | PARTNERS (https://partners.f5.com/) | MYF5 (https://my.f5.com/)

(/s/) TOPICS QUESTIONS (/s/questions) ARTICLES (/s/articles) CODE (/s/codeshare) RESOURCES (/s/resources) ABOUT (/s/getting-started)

F5.COM (https://f5.com/)DEVCENTRAL (/s/)SUPPORT (https://support.f5.com/)PARTNERS (https://partners.f5.com/)MYF5 (https://my.f5.com/)

Login (/DEVC_LoginToCommunity?startURL=//s/articles/what-is-big-ip-24596) | Sign up (/DEVC_SignUpToCommunity) |

Back to Article List

What Is BIG-IP?

Updated 1 year ago | Originally posted January 19, 2017 by Chase Abbott ● F5 (/s/profile/0051T000008twwFQAQ)

Topics in this Article: application delivery (/s/articles?tag=application delivery), big-ip (/s/articles?tag=big-ip), devcentral basics (/s/articles?tag=devcentral basics)

DevCentral**Basics**

tl;dr - BIG-IP is a collection of hardware platforms and software solutions providing services focused on security, reliability, and performance.

F5's BIG-IP is a family of products covering software and hardware designed around application availability, access control, and security solutions. That's right, the BIG-IP name is interchangeable between F5's software and hardware application delivery controller and security products. This is different from BIG-IQ, a suite of management and orchestration tools, and F5 Silverline, F5's SaaS platform. When people refer to BIG-IP this can mean a single software module in BIG-IP's software family or it could mean a hardware chassis sitting in your datacenter. This can sometimes cause a lot of confusion when people say they have question about "BIG-IP" but we'll break it down here to reduce the confusion.

BIG-IP Software

BIG-IP software products are licensed modules that run on top of F5's Traffic Management Operation System® (TMOS). This custom operating system is an event driven operating system designed specifically to inspect network and application traffic and make real-time decisions based on the configurations you provide. The BIG-IP software can run on hardware or can run in virtualized environments. Virtualized systems provide BIG-IP software functionality where hardware implementations are unavailable, including public clouds and various managed infrastructures where rack space is a critical commodity.

BIG-IP Primary Software Modules

- **BIG-IP Local Traffic Manager (LTM)** - Central to F5's full traffic proxy functionality, LTM provides the platform for creating virtual servers, performance, service, protocol, authentication, and security profiles to define and shape your application traffic. Most other modules in the BIG-IP family use LTM as a foundation for enhanced services.
- **BIG-IP DNS** - Formerly Global Traffic Manager, BIG-IP DNS provides similar security and load balancing features that LTM offers but at a global/multi-site scale. BIG-IP DNS offers services to distribute and secure DNS traffic advertising your application namespaces.
- **BIG-IP Access Policy Manager (APM)** - Provides federation, SSO, application access policies, and secure web tunneling. Allow granular access to your various applications, virtualized desktop environments, or just go full VPN tunnel.
- **Secure Web Gateway Services (SWG)** - Paired with APM, SWG enables access policy control for internet usage. You can allow, block, verify and log traffic with APM's access policies allowing flexibility around your acceptable internet and public web application use. You know.... contractors and interns shouldn't use Facebook but you're not going to be responsible why the CFO can't access their cat pics.
- **BIG-IP Application Security Manager (ASM)** - This is F5's web application firewall (WAF) solution. Traditional firewalls and layer 3 protection don't understand the complexities of many web applications. ASM allows you to tailor acceptable and expected application behavior on a per application basis . Zero day, DoS, and click fraud all rely on traditional security device's inability to protect unique application needs; ASM fills the gap between traditional firewall and tailored granular application protection.
- **BIG-IP Advanced Firewall Manager (AFM)** - AFM is designed to reduce the hardware and extra hops required when ADC's are paired with traditional firewalls. Operating at L3/L4, AFM helps protect traffic destined for your data center. Paired with ASM, you can implement protection services at L3 - L7 for a full ADC and Security solution in one box or virtual environment.

BIG-IP Hardware

BIG-IP hardware offers several types of purpose-built custom solutions, all designed in-house by our fantastic engineers; no white boxes here. BIG-IP hardware is offered via series releases, each offering improvements for performance and features determined by customer requirements.

These may include increased port capacity, traffic throughput, CPU performance, FPGA feature functionality for hardware-based scalability, and virtualization capabilities. There are two primary variations of BIG-IP hardware, single chassis design, or VIPRION modular designs. Each offer unique advantages for internal and collocated infrastructures. Updates in processor architecture, FPGA, and interface performance gains are common so we recommend referring to F5's hardware page (https://f5.com/products/deployment-methods/hardware) for more information.

Topics in this Article:

application delivery (/s/articles?tag=application delivery) big-ip (/s/articles?tag=big-ip)

devcentral basics (/s/articles?tag=devcentral basics)

The DevCentral Team (/s/profile/0051T000008OdpPQAS) **(F5)**

published this new Knowledge.

May 15, 2019 at 2:40 AM (/s/feed/0D51T00006j1aGLSAY)

2 comments 66 views

Like Comment

More comments 1 of 2

Leonardo Souza (/s/profile/0051T000008ud3XQAQ)
3 years ago

"There are two primary variations of BIG-IP hardware, single chassis design, or VIPRION modular designs."

I don't think VIPRION is considered to be in the BIG-IP hardware family. However, iSeries is branded as BIG-IP hardware.

Like

Login to comment on this post

👍 0 👎 0

About DevCentral

An F5 Networks Community

We are an online community of technical peers dedicated to learning, exchanging ideas, and solving problems - together.

Learn More (/s/getting-started)

Get a developer Lab license (/s/articles/f5-developer-edition-how-to-obtain-a-developer-license-key)

Contact us - Feedback and Help (/s/contactsupport)

Become an MVP (/s/mvp)

HAVE A QUESTION? | Support and Sales > (https://www.f5.com/company/contact)

Follow Us (https://twitter.com/f5networks)
(https://www.linkedin.com/company/f5-networks/)
(https://www.facebook.com/f5networksinc)
(https://www.youtube.com/user/f5networksinc) (/s/)

About F5

Corporate Information
(https://www.f5.com/company)
Newsroom
(https://www.f5.com/company/news)
Investor Relations
(https://www.f5.com/company/investor-relations)
Careers
(https://f5.com/careers)
Contact Information
(https://f5.com/about-us/contact)
Communication Preferences
(https://interact.f5.com/F5-Preference-Center.html?utm_source=optin-f5footer)
Product Certifications
(https://www.f5.com/company/certifications)

Education

Training
(https://www.f5.com/services/training)

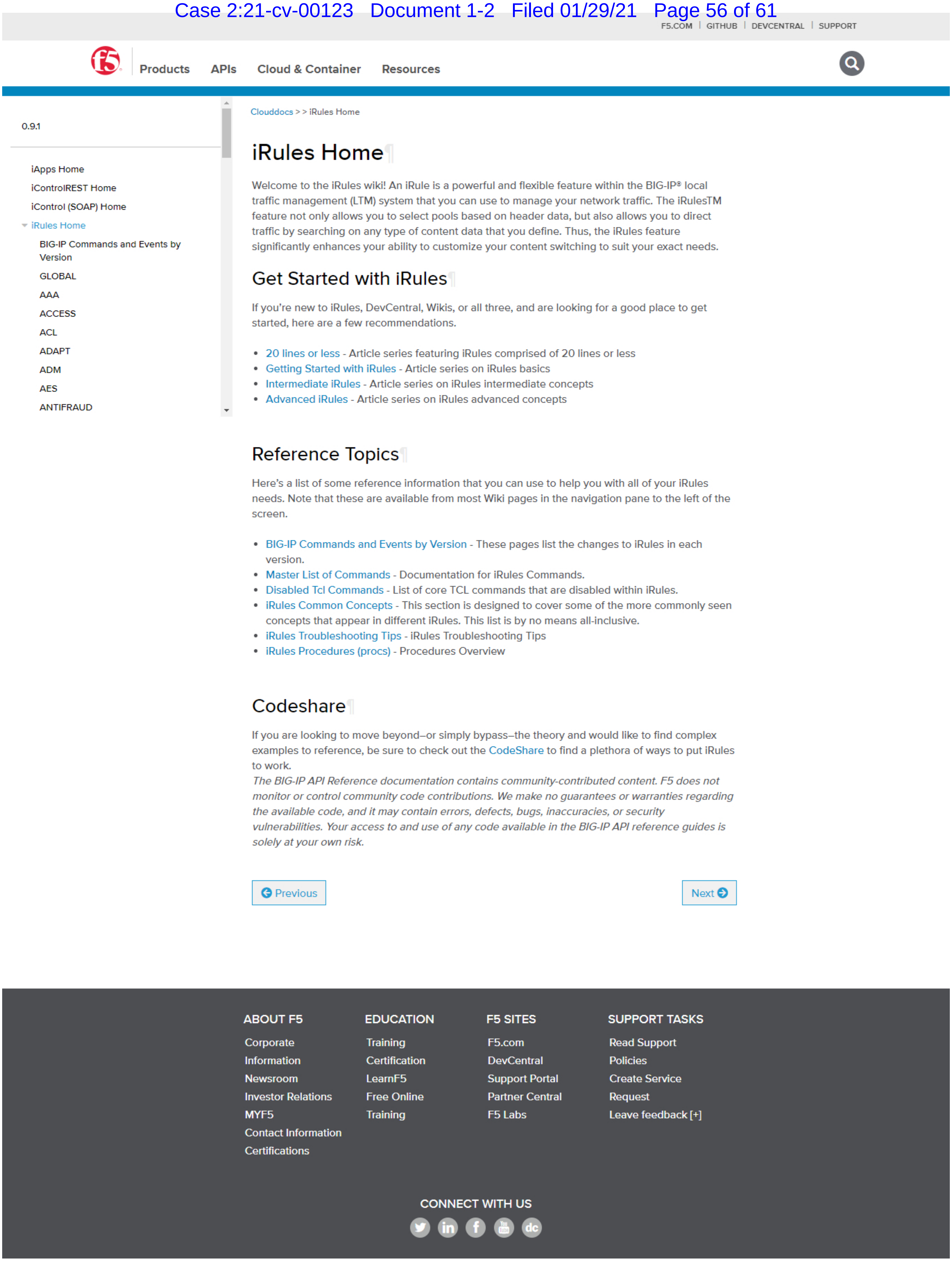
Professional Certification
(https://www.f5.com/services/certification)
LearnF5
(https://account.f5.com/learnf5/signin)
Free Online Training
(https://f5.com/education/training/free-courses)

F5 Sites

F5.com
(https://www.f5.com/)
DevCentral
(/s)
Support Portal
(https://support.f5.com/csp/home)
Partner Central
(https://partners.f5.com/)
F5 Labs
(https://www.f5.com/labs)

Policies (https://www.f5.com/company/policies) | Privacy (https://www.f5.com/company/policies/privacy-notice) | Trademarks (https://www.f5.com/company/policies/trademarks) | | DevCentral Terms of Use (/s/EULA)

Exhibit E



0.9.1

- iApps Home
- iControlREST Home
- iControl (SOAP) Home
- iRules Home
  - BIG-IP Commands and Events by Version
  - GLOBAL
  - AAA
  - ACCESS
  - ACL
  - ADAPT
  - ADM
  - AES
  - ANTIFRAUD

Clouddocs > > iRules Home

iRules Home

Welcome to the iRules wiki! An iRule is a powerful and flexible feature within the BIG-IP® local traffic management (LTM) system that you can use to manage your network traffic. The iRulesTM feature not only allows you to select pools based on header data, but also allows you to direct traffic by searching on any type of content data that you define. Thus, the iRules feature significantly enhances your ability to customize your content switching to suit your exact needs.

Get Started with iRules

If you're new to iRules, DevCentral, Wikis, or all three, and are looking for a good place to get started, here are a few recommendations.

- 20 lines or less - Article series featuring iRules comprised of 20 lines or less
- Getting Started with iRules - Article series on iRules basics
- Intermediate iRules - Article series on iRules intermediate concepts
- Advanced iRules - Article series on iRules advanced concepts

Reference Topics

Here's a list of some reference information that you can use to help you with all of your iRules needs. Note that these are available from most Wiki pages in the navigation pane to the left of the screen.

- BIG-IP Commands and Events by Version - These pages list the changes to iRules in each version.
- Master List of Commands - Documentation for iRules Commands.
- Disabled Tcl Commands - List of core TCL commands that are disabled within iRules.
- iRules Common Concepts - This section is designed to cover some of the more commonly seen concepts that appear in different iRules. This list is by no means all-inclusive.
- iRules Troubleshooting Tips - iRules Troubleshooting Tips
- iRules Procedures (procs) - Procedures Overview

Codeshare

If you are looking to move beyond–or simply bypass–the theory and would like to find complex examples to reference, be sure to check out the CodeShare to find a plethora of ways to put iRules to work.
The BIG-IP API Reference documentation contains community-contributed content. F5 does not monitor or control community code contributions. We make no guarantees or warranties regarding the available code, and it may contain errors, defects, bugs, inaccuracies, or security vulnerabilities. Your access to and use of any code available in the BIG-IP API reference guides is solely at your own risk.

ABOUT F5
- Corporate Information
- Newsroom
- Investor Relations
- MYF5
- Contact Information
- Certifications

EDUCATION
- Training
- Certification
- LearnF5
- Free Online Training

F5 SITES
- F5.com
- DevCentral
- Support Portal
- Partner Central
- F5 Labs

SUPPORT TASKS
- Read Support Policies
- Create Service Request
- Leave feedback [+]

CONNECT WITH US

Exhibit F

AskF5 Home / Knowledge Centers / BIG-IP AAM / BIG-IP Acceleration: Concepts / Managing Traffic with Rate Shaping

Manual Chapter : Managing Traffic with Rate Shaping

Applies To:

Show Versions ⊞

Table of Contents | << Previous Chapter | Next Chapter >>

Managing Traffic with Rate Shaping

Introduction to rate shaping

The BIG-IP® system includes a feature called rate shaping. ***Rate shaping*** allows you to enforce a throughput policy on incoming traffic. Throughput policies are useful for prioritizing and restricting bandwidth on selected traffic patterns.

Rate shaping can be useful for an e-commerce site that has preferred clients. For example, the site might want to offer higher throughput for preferred customers, and lower throughput for other site traffic.

The rate shaping feature works by first queuing selected packets under a rate class, and then dequeuing the packets at the indicated rate and in the indicated order specified by the rate class. A ***rate class*** is a rate-shaping policy that defines throughput limitations and a packet scheduling method to be applied to all traffic handled by the rate class.

You configure rate shaping by creating one or more rate classes and then assigning the rate class to a packet filter or to a virtual server. You can also use the iRules® feature to instruct the BIG-IP system to apply a rate class to a particular connection.

You can apply a rate class specifically to traffic from a server to a client or from a client to a server. If you configure the rate class for traffic that is going to a client, the BIG-IP system does not apply the throughput policy to traffic destined for the server. Conversely, if you configure the rate class for traffic that is going to a server, the BIG-IP system does not apply the throughput policy to traffic destined for the client.

About rate classes

A rate class defines the throughput limitations and packet scheduling method that you want the BIG-IP® system to apply to all traffic that the rate class handles. You assign rate classes to virtual servers and packet filter rules, as well as through iRules®.

If the same traffic is subject to rate classes that you have assigned from more than one location, the BIG-IP system applies the last-assigned rate class only. The BIG-IP system applies rate classes in the following order:

- The first rate class that the BIG-IP system assigns is from the last packet filter rule that matched the traffic and specified a rate class.
- The next rate class that the BIG-IP system assigns is from the virtual server; if the virtual server specifies a rate class, the rate class overrides any rate class that the packet filter selects.
- The last rate class assigned is from the iRule; if the iRule specifies a rate class, this rate class overrides any previously-selected rate class.

Note: *Rate classes cannot reside in partitions. Therefore, a user's ability to create and manage rate classes is defined by user role, rather than partition-access assignment.*

You can create a rate class using the BIG-IP Configuration utility. After you have created a rate class, you must assign it to a virtual server or a packet filter rule, or you must specify the rate class from within an iRule.

Rate class name

The first setting you configure for a rate class is the rate class name. Rate class names are case-sensitive and might contain letters, numbers, and underscores (_) only. Reserved keywords are not allowed.

Each rate class that you define must have a unique name. This setting is required.

To specify a rate class name, locate the `Name` field on the New Rate Class screen and type a unique name for the rate class.

Base rate

The **Base Rate** setting specifies the base throughput rate allowed for traffic that the rate class handles. The sum of the base rates of all child rate classes attached to a parent rate class, plus the base rate of the parent rate class, cannot exceed the ceiling of the parent rate class. For this reason, F5 Networks® recommends that you always set the base rate of a parent rate class to 0 (the default value).

You can specify the base rate in bits per second (bps), kilobits per second (Kbps), megabits per second (Mbps), or gigabits per second (Gbps). The default unit is bits per second. This setting is required.

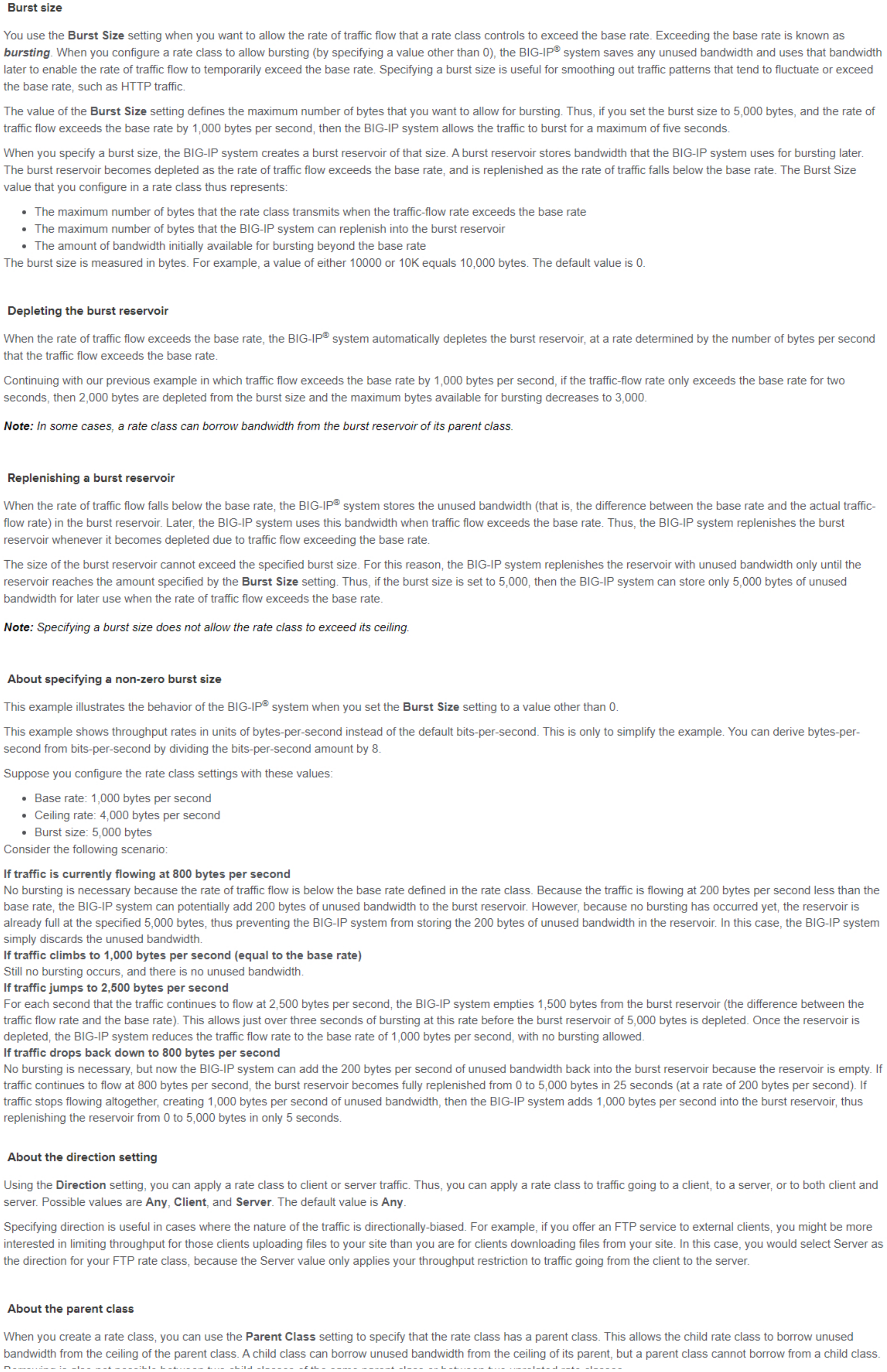
Note: *These numbers are powers of 10, not powers of 2.*

Ceiling rate

The **Ceiling Rate** setting specifies the absolute limit at which traffic is allowed to flow when bursting or borrowing. You can specify the ceiling in bits per second (bps), kilobits per second (Kbps), megabits per second (Mbps), or gigabits per second (Gbps). The default unit is bits per second.

If the rate class is a parent rate class, the value of the ceiling defines the maximum rate allowed for the sum of the base rates of all child rate classes attached to the parent rate class, plus the base rate of the parent rate class.

Note: *A child rate class can borrow from the ceiling of its parent rate class.*



Burst size

You use the **Burst Size** setting when you want to allow the rate of traffic flow that a rate class controls to exceed the base rate. Exceeding the base rate is known as ***bursting***. When you configure a rate class to allow bursting (by specifying a value other than 0), the BIG-IP® system saves any unused bandwidth and uses that bandwidth later to enable the rate of traffic flow to temporarily exceed the base rate. Specifying a burst size is useful for smoothing out traffic patterns that tend to fluctuate or exceed the base rate, such as HTTP traffic.

The value of the **Burst Size** setting defines the maximum number of bytes that you want to allow for bursting. Thus, if you set the burst size to 5,000 bytes, and the rate of traffic flow exceeds the base rate by 1,000 bytes per second, then the BIG-IP system allows the traffic to burst for a maximum of five seconds.

When you specify a burst size, the BIG-IP system creates a burst reservoir of that size. A burst reservoir stores bandwidth that the BIG-IP system uses for bursting later. The burst reservoir becomes depleted as the rate of traffic flow exceeds the base rate, and is replenished as the rate of traffic falls below the base rate. The Burst Size value that you configure in a rate class thus represents:

- The maximum number of bytes that the rate class transmits when the traffic-flow rate exceeds the base rate
- The maximum number of bytes that the BIG-IP system can replenish into the burst reservoir
- The amount of bandwidth initially available for bursting beyond the base rate

The burst size is measured in bytes. For example, a value of either 10000 or 10K equals 10,000 bytes. The default value is 0.

Depleting the burst reservoir

When the rate of traffic flow exceeds the base rate, the BIG-IP® system automatically depletes the burst reservoir, at a rate determined by the number of bytes per second that the traffic flow exceeds the base rate.

Continuing with our previous example in which traffic flow exceeds the base rate by 1,000 bytes per second, if the traffic-flow rate only exceeds the base rate for two seconds, then 2,000 bytes are depleted from the burst size and the maximum bytes available for bursting decreases to 3,000.

Note: *In some cases, a rate class can borrow bandwidth from the burst reservoir of its parent class.*

Replenishing a burst reservoir

When the rate of traffic flow falls below the base rate, the BIG-IP® system stores the unused bandwidth (that is, the difference between the base rate and the actual traffic-flow rate) in the burst reservoir. Later, the BIG-IP system uses this bandwidth when traffic flow exceeds the base rate. Thus, the BIG-IP system replenishes the burst reservoir whenever it becomes depleted due to traffic flow exceeding the base rate.

The size of the burst reservoir cannot exceed the specified burst size. For this reason, the BIG-IP system replenishes the reservoir with unused bandwidth only until the reservoir reaches the amount specified by the **Burst Size** setting. Thus, if the burst size is set to 5,000, then the BIG-IP system can store only 5,000 bytes of unused bandwidth for later use when the rate of traffic flow exceeds the base rate.

Note: *Specifying a burst size does not allow the rate class to exceed its ceiling.*

About specifying a non-zero burst size

This example illustrates the behavior of the BIG-IP® system when you set the **Burst Size** setting to a value other than 0.

This example shows throughput rates in units of bytes-per-second instead of the default bits-per-second. This is only to simplify the example. You can derive bytes-per-second from bits-per-second by dividing the bits-per-second amount by 8.

Suppose you configure the rate class settings with these values:

- Base rate: 1,000 bytes per second
- Ceiling rate: 4,000 bytes per second
- Burst size: 5,000 bytes

Consider the following scenario:

If traffic is currently flowing at 800 bytes per second
No bursting is necessary because the rate of traffic flow is below the base rate defined in the rate class. Because the traffic is flowing at 200 bytes per second less than the base rate, the BIG-IP system can potentially add 200 bytes of unused bandwidth to the burst reservoir. However, because no bursting has occurred yet, the reservoir is already full at the specified 5,000 bytes, thus preventing the BIG-IP system from storing the 200 bytes of unused bandwidth in the reservoir. In this case, the BIG-IP system simply discards the unused bandwidth.
If traffic climbs to 1,000 bytes per second (equal to the base rate)
Still no bursting occurs, and there is no unused bandwidth.
If traffic jumps to 2,500 bytes per second
For each second that the traffic continues to flow at 2,500 bytes per second, the BIG-IP system empties 1,500 bytes from the burst reservoir (the difference between the traffic flow rate and the base rate). This allows just over three seconds of bursting at this rate before the burst reservoir of 5,000 bytes is depleted. Once the reservoir is depleted, the BIG-IP system reduces the traffic flow rate to the base rate of 1,000 bytes per second, with no bursting allowed.
If traffic drops back down to 800 bytes per second
No bursting is necessary, but now the BIG-IP system can add the 200 bytes per second of unused bandwidth back into the burst reservoir because the reservoir is empty. If traffic continues to flow at 800 bytes per second, the burst reservoir becomes fully replenished from 0 to 5,000 bytes in 25 seconds (at a rate of 200 bytes per second). If traffic stops flowing altogether, creating 1,000 bytes per second of unused bandwidth, then the BIG-IP system adds 1,000 bytes per second into the burst reservoir, thus replenishing the reservoir from 0 to 5,000 bytes in only 5 seconds.

About the direction setting

Using the **Direction** setting, you can apply a rate class to client or server traffic. Thus, you can apply a rate class to traffic going to a client, to a server, or to both client and server. Possible values are **Any**, **Client**, and **Server**. The default value is **Any**.

Specifying direction is useful in cases where the nature of the traffic is directionally-biased. For example, if you offer an FTP service to external clients, you might be more interested in limiting throughput for those clients uploading files to your site than you are for clients downloading files from your site. In this case, you would select Server as the direction for your FTP rate class, because the Server value only applies your throughput restriction to traffic going from the client to the server.

About the parent class

When you create a rate class, you can use the **Parent Class** setting to specify that the rate class has a parent class. This allows the child rate class to borrow unused bandwidth from the ceiling of the parent class. A child class can borrow unused bandwidth from the ceiling of its parent, but a parent class cannot borrow from a child class. Borrowing is also not possible between two child classes of the same parent class or between two unrelated rate classes.

Borrowing is also not possible between two child classes of the same parent class or between two unrelated rate classes.

A parent class can itself have a parent, provided that you do not create a circular dependency. A ***circular dependency*** is a relationship where a rate class is a child of itself, directly or indirectly.

If a rate class has a parent class, the child class can take unused bandwidth from the ceiling of the parent class. The process occurs in this way:

- If the rate of traffic flow to which the child class is applied exceeds its base rate, the child class begins to deplete its burst reservoir as described previously.
- If the reservoir is empty (or no burst size is defined for the rate class), then the BIG-IP® system takes unused base-rate bandwidth from the ceiling of the parent class and gives it to the child class.
- If the unused bandwidth from the parent class is depleted, then the child class begins to use the reservoir of the parent class.
- If the reservoir of the parent class is empty (or no burst size is defined for the parent class), then the child class attempts to borrow bandwidth from the parent of the parent class, if the parent class has a parent class.
- This process continues until there is no remaining bandwidth to borrow or there is no parent from which to borrow.

Borrowing only allows the child to extend its burst duration; the child class cannot exceed the ceiling under any circumstance.

Note: *Although the above description uses the term "borrowing", bandwidth that a child class borrows is not paid back to the parent class later, nor is unused bandwidth of a child class returned to its parent class.*

About shaping policy

This setting specifies a shaping policy that includes customized values for drop policy and queue method. The default value is None.

You can create additional shaping policies using the Traffic Management shell (tmsh).

About queue method

The **Queue Method** setting determines the method and order in which the BIG-IP® system dequeues packets.

A rate class supports two queue methods:

Stochastic Fair Queue
Stochastic Fair Queueing (***SFQ***) is a queuing method that queues traffic under a set of many lists, choosing the specific list based on a periodically-changing hash of the connection information. This results in traffic from the same connection always being queued in the same list. SFQ then dequeues traffic from the set of the lists in a round-robin fashion. The overall effect is that fairness of dequeuing is achieved because one high-speed connection cannot monopolize the queue at the expense of slower connections.
Priority FIFO
The ***Priority FIFO*** (***PFIFO***) queuing method queues all traffic under a set of five lists based on the Type of Service (ToS) field of the traffic. Four of the lists correspond to the four possible ToS values (Minimum delay, Maximum throughput, Maximum reliability, and Minimum cost). The fifth list represents traffic with no ToS value. The PFIFO method then processes these five lists in a way that attempts to preserve the meaning of the ToS field as much as possible. For example, a packet with the ToS field set to Minimum cost might yield dequeuing to a packet with the ToS field set to Minimum delay.

About drop policy

The BIG-IP® system drops packets whenever the specified rate limit is exceeded. A drop policy specifies the way that you want the system to drop packets. The default value is **fred**.

Note: *You cannot use* ***fred*** *or* ***red****, if you select* ***sfq*** *for the* ***Queue Method*** *setting.*

Possible values are:

fred
Specifies that the system uses Flow-based Random Early Detection to determine whether to drop packets, based on the aggressiveness of each flow. If you require flow fairness across the rate class, select **fred**.
red
Specifies that the system randomly drops packets.
tail
Specifies that the system drops the end of the traffic stream.
You can create additional drop policies using the Traffic Management shell (tmsh).

Table of Contents | << Previous Chapter | Next Chapter >>